Specijalno izdanje časopisa „Galaksija“ Jul 1984. / Cena 200 D

računari u vašoj kući 2

škola akcionih igara

memorija za „galaksiju“ 48 K

QL • Electron • BBC B • Commodore 64




БЕОБАНКА
свуда са вама

računari
u vašoj kući

2

Specijalno izdanje časopisa „Galaksija"
Jul 1984.
Cena 200 D

Sadržaj

4/ni golub na grani, ni vrabac na ruci

6/na zapadu ništa novo

8/u susret superinteligenciji

11/era tehnoloških genija

16/nemojte plakati za epiom

18/računari u izlogu. QL, Elektron, BBC B, Commodore 64

32/štampači u akciji

42/moj prvi program

47/biblioteka programa

55/„galaksija" bez tajni

66/svi „spectrumovi" restarti

68/softverska veza

72/škola akcionih igara

82/programator epproma

88/memorija za „galaksiju"

94/„ubiću se, neće da radi!"

96/konstruktorski hod po mukama

Na naslovnoj strani:
CRTEŽ SA KORICA KNJIGE THE JOY OF COMPUTERS KOJA ĆE POD NASLOVOM KOMPJUTOR U KUĆI IZAĆI OVE JESENI NA HRVATSKOSRPSKOM ODNOSNO SLOVENAČKOM JEZIKU U IZDANJU ČANKARJEVE ZALOŽBE (videti oglas na str. 38)

Izdaje
Beogradski izdavačko-grafički zavod
OOUR Novinske delatnosti „Duga"
11000 Beograd
Bulevar vojvode Mišića 17

Telefoni
650-161 (redakcija)
650-528 (prodaja)
651-793 (propegenda)

Generalni direktor
Gojko Žečer

Direktor OOUR „Duga"
Zoran Milošević

Glavni i odgovorni urednik
Gavrilo Vučković

Urednik izdanja
Jova Regasek

Likovna i grafička oprema
Dušan Mijetović

Redakcija časopisa „Galaksija"
Tanasije Gavranović pomoćnik glavnog i odgovornog urednika
Esad Jakupović zamenik glavnog i odgovornog urednika
Aleksandar Milinković urednik
Jova Regasak urednik
Zorka Simović sekretar redakcije
Gavrilo Vučković glavni i odgovorni urednik

Autori lakstava
Dejan Risanović
Miša Hadži Đorđević
Ivan Garenčir
Nadeljko Perazenović
Milan Pavićavić
Borivoje Perzić
Jova Regasak
Slobodan Vujović
Andelko Zgorelec

Prevodioci
Domagoj Božić
Zdenko Dizdar
Gavrilo Vučković

Tehnička saradnja
Ljubiša Milovanović
Mira Tiodorović

Specijalna saradnja
Andelko Zgorelac London

Izdavački savet „Galaksije"
Dr Rudi Dobnađi, prof. dr Branislav Dimitrijević (predsednik) Radovan Draškić Tanasije Gavranović Živorad Gligić Esad Jakupović Valizar Masleć Nikola Pejić Željko Perunović prof. dr Momčilo Ristić Vlada Ristić dr inž. Milorad Teofilović Vidojko Veličković Valimir Vasović Milivoje Vuković

Štampa
Beogradski izdavačko-grafički zavod
11000 Beograd, Bulevar vojvode Mišića 17

Žiro-račun kod SDK
60802-603-17132
Devizni račun kod Beobanke
60811-620-5-82701 999-01066
Za inostranstvo cena dvostruka
(400 D, 3 US $, 9 DM, 55 Sch, 7 Sfrs, 24 Ffrs, 2 Lstg)
Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i Službenog glasnika broj 26/72. ovo izdanje oslobođeno je poreza na promet

Zašto kasnimo?

ni golub na grani ni vrabac u ruci

Kada smo pre nepunih godinu dana u redakciji podelili da razmišljamo o jednom posebnom izdanju koje bi u celini bilo posvećeno kućnim računarima nismo ni slutili šta se sve nalazi na drugom kraju fitilja koji smo time nameravali da potpalimo. Računari u našoj kući – koji su se pojavili na tržištu prvih dana januara planuli su u tiražu od 50 000 primeraka a male „galaksije" izrasle je preko noći u naš nacionalni računar i na samo to. Računari već mesecima na silaze sa stranica štampe i programa radija i televizije a pored nekoliko računarskih naslova koji sa već nalaze u izlozima knjižara izdavači najavljuju još silaset novih. Slovenci su pokrenuli već dva specijalizovana časopisa a sasvim određena planove ima i redakcija „Galaksije".

Računarski izazov probudio je bar em ra rečima i domaću elektronsku industriju koja do pre nekoliko meseci smatrajući da smo previše siromašni za to o kućnim računarima nije želela čak ni da razmišlja. Uz „galaksiju" i „lolu 8" i veražlinskog „galeba" originalna računara najavljuju i Elektronska industrija Iskra-Delta i Slovenijales. Na domaća kompjuterska komešanja nema odziva jedino iz SIV-a koji prekićeno kućne računare još uvek drži pod zabranom. To je razlog više da domaćoj industriji čvrsto silsnemo palčeve. Računarsku revoluciju uostalom možemo ostvariti samo ako budemo imali domaći računar.

I sama redakcija je ruku na srce pomalo izgorela na vatri koju je zapalile sopstvenim rukama. Akcija sa računarom „galaksija" uvela je u redakciju vanredno stanje koje traje već mesecima. Epromi instalirani štampana kola čipovi neprekidno odvlače našu pažnju sa svakodnevnih poslova. To je malo usporilo pripreme „Računara 2". Ne ovom specijalnom izdanju osim toga radi veoma skromna ekipe ljudi — mala redakcija i mala grupa spoljnih saradnika. Zbog toga nismo bili u stanju da se njim izađemo pred čitaoce onda kada smo to želeli ali smo redakcijske poslove okončali znatno ranije nego što je naša štamparija zbog nedostatka papira bila u stanju to da prihvati. To nravno na znači da pred vas izlazimo sa bajatim tekstovima. Većina među njima ima trajnu vrednost i oni kod kojih je aktuelnost posebno važna dopunjavani su do poslednjeg dana. Članak o računaru QL napisan je ponovo nakon što su u redakciju prispeli prvi realni prikazi neposredno pred ulazak izdanja u štampu.

Čitav taj neplanirani vremenski pomak uslovio je i izvesne promena u već najavljenoj koncepciji specijalnog izdanja. U obilju ideja i materijala i na ipak ograničenom prostoru pojedine nagoveštane rubrike i priloge morali smo da odložimo za sledeći broj „Računara". Izostao je između ostalog i najavljeni umetak taj prostor sa standardnim prelomom ustupili smo izvesnom broju tekstova koji će verujemo više odgovarati ukusu većina. Dodatni razlog za sve ove promena bio je i povećan priliv oglasa za koje se nadamo da nikako nisu nužno zlo nego specifična informacije čiji značaj nije za podcenjivanje.

Ako ovim izdanjem nismo u svemu zadovoljili sveži ukus molimo vas za malo strpljenja pripreme „Računara 3" već su uvelike u toku (nastojaćemo da iz štampa izađu krajem septembra) i u njima ćete verujemo pored ostalog naći i one priloge koji su po vašem mišljenju trebalo da nađu svoje mesto u publikaciji koju upravo čitate.

Redakcija

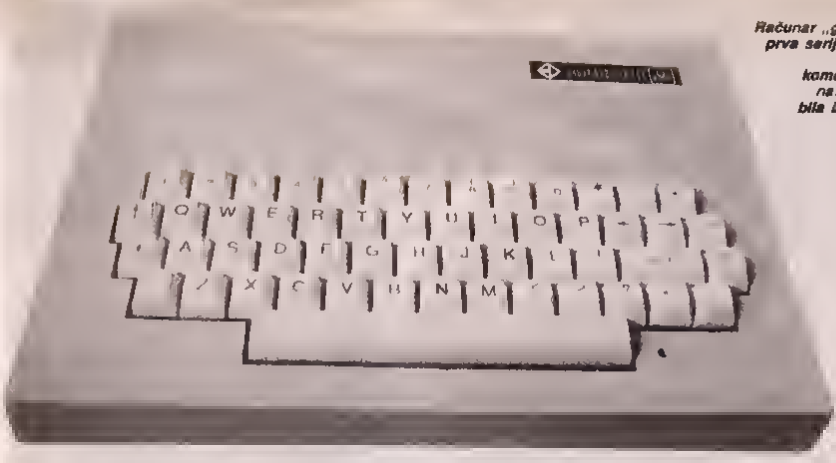
Ovogodišnji Sajam tehnika u Beogradu, koji je trajao od 22. do 26. maja, protekao je, više nego ijedan pre njega, u znaku kućnih i, naročito, malih poslovnih računara. U hali 1, na ogromnom iziožbenom prostoru, mogio je da se pogiada i proba obilje domaćih i stranih mini i mikroračunara, kojih, na žalost, i dalje nama u prodavnicama. Sa posebnim uzbuđanjem smo očekivali da vidimo da li je (i kako) domaća elektronska industrija reagovala na „provokaciju" zvanu računar „galaksija". U nezvaničnim razgovorima sa zvaničnim licima na štandovima pokušali smo da saznamo nešto više o planovima domaćih proizvođača računara i modelima u koja se ulažu najveće nade.

Jugoslovenski računari nije bilo teško zaključiti nisu u pravom smislu kućni. Dimanzije su im doduše najčešće takve da sasvim lepo mogu da se smeste na vaš radni sto, mogućnosti dovoljne za sve (pogotovu kućne!) primene ali sve to po ceni od koje vas odmah zaboli glava. Naši proizvođači su dakle najviše okrenuti firmama koje bi želele da se kompjuterizuju plećajući ovaj tehnološki napredak u dinarime maloprodaja interesuje sliebo koga (za maloprodaju naravno, trabaju i posebna dozvole) i to najpre zato što je u našim uslovima teško realizovati veliku i jeftinu seriju.

Šta sprema Ei

Obilazeći halu 1 u smeru kretenja kazaljke na časovniku naišli emo najpre na štand Toverne Meril. Ona prodaje poznate Kopa terminale koji su u poslednje vreme posiali standard u mnogim računskim centrima. Ništa za nas, rekio bi se. Ali na štendu je izložen i domeći računar PMP koga se ne bismo postidali ni na svetskom tržištu. Imeli smo zadovoljstvo da porazgovorerno sa Marijanom Miletićem koji je zajedno sa kolegom Brankom Javtićem konstruisao PMP računar koji kako ćemo videli ima najmoćniju programaku podršku od svih kućnih modela. Ovaj kompjuter je naime zasnovan na mikroprocesoru TCT 1t koji predstevlja PDP 11 računar u malom. Operativni sistem je tekav da svi programi pisani za PDP računare bez ikakvih problema rade i na PMP-ju. A poznato je da PDP računare ima ogroman broj, da se konsle bezmalo u svim računskim centrima u svetu kao centralne ili periferijske jedinice i da kako reče Marijen Milelić on ličnu pokiadja jedan primerak svoga računare onome ko navede jedan programski jezik koji im nije prilagođeni ROM računara PMP ime svega 2 KB i u njega je smašten samo kreiki program koji učiteva u RAM operativni sistem sa jedna od disketa. RAM u osnovnoj konfiguraciji ima 64 kilobajta ali sa obzirom da je TCT 11 šesnaestobitni procesor može praktično neograničeno proširivati.

Kako stoji stvar sa cenom? Centralna jedinica računara košte 150 hiljada dinara a svake disk-jedinica po sto. Složićete se sa nama da ova cena, kada se uzme u obzir ono što se dobije nije velike i da ni sireni računari uz carine ne bi košteli manje u odgovarajućoj konfiguraciji. Na žalost postoji jedno eli PMP ne može da se koristi bez terminala a terminal zavisno od mogućnosti može da košta 30—70 sienih milional Utešie nas je jedino tvrdnja Tovarne Meril da rade na integralnom video delu za televizor i posebnoj tastaturi koji će ukoliko ispitivenje ižrište koje je u toku da povoljne rezultate učinili PMP pristupačnim širim krugovima korisnike.



Računar „galaksija" - iako je prva serija od 300 komada privedena kraju, komercijalna varijanta našeg računara nije bila izložena na sajmu

Na obrativši možda i neopravdano dovoljno pažnje Digitronovom štandu stigli smo do sledećeg izlagača — El-Honeywell. Naša žurba nije bila neobična — Politika je nedavno donela vest da će El uskoro izbaciti na tržište prvi (svi koristimo taj epitet!) kućni računar prilagođen svakome. Čim smo videli da na štandu nema gužve znali smo da nam njemu nema ni tog računara. Tačno tako — El je izložio već dobro poznati Honeywell 6/10. Ovaj računar je zasnovan na Honeywell-ovom mikroprocesoru LSI 6 koji se radi kompatibilnosti može dopuniti čuvenim Intelovim 8086. Korisničke memorije od 128 kilobajta može da se proširuje do 512 Kb dok su na asinhroni komunikacioni port priključena dve disk-jedinice po 650 Kb. Oni koji žele da obrađuju veće količine informacija mogu da dodaju Winchester masivna diskova po 20 Mb. Štampač se naknadno dokupljuje i može biti 132-kolonski matrični (100 karaktera u sekundi) ili sa tepezom — 30 karaktera u sekundi. Za cenu ovakve konfiguracije se nismo raspitivali — smatrali smo da vas neće interesovati.

Od „galeba" do „orla"

Na sledećem štandu smo ugledali dobro poznati računar (Apple II) i dobro poznatog kolegu (Julija Makaneca mlađeg). Došli smo jasno do štanda Velebita. Čitaocima „Galaksije" je svakako, poznato da su Varaždinci pre tačno godinu dana („Galaksija" broj 135) izbacili na tržišta seriju od 40 računara „galeb". Ovaj računar zbog predugog razvoja i unekoliko zastarele koncepcije koju je dopunila veoma visoka cena 90 000 dinara nije zaživeo na tržištu i, po svemu sudeći, pripada prošlosti. No, „galebov" naslednik koji se zove „orao" treba uskoro da uzleti i zauzme mesto na našim stolovima. „Orao" je logičan nastavak „galeba" i delo je istog konstruktora — Mire Kocijana. Zasnovan je na mikroprocesoru 6502 i poseduje ROM od 16 i RAM od 6 Kb koji može dalje da proširuje. (Sve ove informacije su dobijene u nezvaničnom razgovoru što znači da neka od njih mogu i da se promene). Kao periferijska jedinica koristi se standardni crno-beli televizor na kome računar kontroliše tinu grafiku sa 256×128 tačaka i kasetofon ali postoji i RS 232 port za dalja proširenja a u računar može da se ugradi i disk-interfejs. Bejzik bi obzirom na pristojan kapacitet ROM-a trebao da bude vrlo dobar i dopunjen grafičkim rutinama (crtanje krugova itd.) ali kako izgleda, ne i asemblerom koji bi omogućio jednostavno pisanje mašinskih programa.

„Orao" će izgleda koštati koliko i „galeb" — oko 9 starih miliona. Može li kao takav, da konkuriše Spectrumu koji se prodaje preko malih oglasa i, za duplo manje novca, nudi enormno programsku podršku? Na to pitanje mi je teško odgovoriti kakav god bio, „orao" je ipak naš računar. Velebit jasno pokazuje da smatra da je njegov novi računar maksimalno konkurentan — izrađeno je 1000 štampanih ploča što je za naše prilike ogromna serija. Međutim, na sajamskom štandu još nema prototipa ni prospekata izgleda da razvoj „orla" (pogotovu njegovog softvera) još nije završen.

Očekujući „orla", ostalo nam je samo da pogledamo Apple II+ računare. Velebit će već u trećem kvartalu ova godina prodavati poznati Apple IIe, koji će, sa jednom tlopi-disk jedinicom monitorom i štampačem Epson MX 100 ili FX 80 koštati nešto manje od 100 starih miliona. S obzirom na moćnu programsku podršku koju Apple ima, čini nam se da za mnoge škola predstavlja prilično zanimljivu alternativu.

„Partner" i „Iola 8"

Iskra Delta je i ove godine izložila poznati Partner. Radi se o poslovnom sistemu koji je zasnovan na mikroprocesoru Z80A, opremljen sa 128 Kb korisnička (RAM) memorija, jednom disk jedinicom od 1 Mb i Winchestar masivnum diskom od 10 Mb. Tastatura je odvojena od centralne jedinice (veza se ostvaruje prako kabla) na kojoj se nalazi i crno-beli monitor koji omogućava ispisivanje 00 znakova u svakom od 24 reda. Pošto je ostvarena CP/M kompatibilnost Partnerovim korisnicima su na raspolaganju mnogobrojni programski jezici ali izgleda da su mnoge firme zainteresovane samo za računar koji će izvršavati određena knjigovodstvene poslove. Za njih se Partner isporučuje bez programskog jezika i opremljen samo potrebnim programom. Cena računara u standardnoj konfiguraciji koja uračunava i TRS štampač i jedan programski jezik iznosi 330 starih miliona.

Završavajući krug, stigli smo do kompjuterskog štanda „Iva Lola Ribara" sa desetak računara Iola 8 na kojima su zainteresovani mogli slobodno da rade. Lola 8 je prisutan već dve godina i polako (nadajmo se i sigurno) krči put ka tržištu. Operativni sistem računara je definitivno završen. Određen broj primeraka Iola 8 je već našao svoje mesto u školama, a njima se pridružio i računar koji su na ovogodišnjem republičkom takmičenju u programiranju osvojili učenici beogradske Matematičke gimnazije „Veljko Vlahović". Na sajmu je održano i Savezno takmičenje koje se nije baš slavno završilo: učestvovale su tri ekipe (Beograd, Novi Sad i Zagreb) kojima su nadležni na sajmu u 18 časova jednostavno isključili struju i tako obrisali programe koji su trebali da budu pisani još čitav čas!

Tri nova domaća kućna računara „galeb", „galaksije" i PMP — predstavljaju baz sumnje bogatu konstruktorsku žetvu. Nova „prva domaća" računara najavljuju i Elektronska industrija, Iskra Delta i Slovenijales. Bilo bi nam drago da se porodica YU računara još više proširi — možda ćemo po broju modela uspeti da prestignemo Engleze — ali bismo iznad svega, voleli da sa barem jedan od njih pojavi u knjižarama robnim kućama i, razume se, našim domovima. Plašimo se da i ove godine sa padom zavese na sajamsku priredbu, na zaboravimo i na sve sjajne planove do sledećeg — sajma.

Dejan Ristanović

5/ni golub na grani, ni vrabac u ruci

Pismo iz Slovenije

SLOVENAČKI RAČUNARSKI BUM

Ono što se sada dešave u Sloveniji mogli bismo da okvalifikujemo kao „kompjuterski bum". Prošlo je više od godinu dana od kako je slovenački mesečnik SUPER-PIP počeo da objavljuje prve skromne priloge sa područja savremena elektronike. Ali svat se menja, potrebe rastu, a i želje mladih za kompjuterskim znanjem sve su veće Zato nije slučajnost što su tom problematikom počeli da se beve i drugi slovenački listovi i revije

Neuobičajeno u svemu tome je što su se ovih dana na slovenačkom novinskom izrštu koje i nije osobito veliko pojevila jedna za drugom dva računarske revije BIT koji će od sada izlaziti jedanput mesečno a izdaje ga Ljubljanski dnevnik i posebno Telekasvo izdanja MOJ MIKRO Tačku na i dodali su Radio študent i ZOTK (Savez organizacija ze tehničku kulturu) Slovenija sa PROGRAMSKOM KASETOM na kojoj sa nalazi 10 programa za kućni računar Sinclair ZX SPECTRUM. Programi su isključivo rad domaćih autora koji su se predstavili javnosti 6 juna ove godine na konferenciji za štampu u Cankarevom domu u Ljubljani

Mladi su to momci mladi kao što je mlad i razvoj računarstva kod nas i zato mislim da će biti pošteno da navedem i njihova imena Žiga Turk Matavž Kmat Darko Vovk Tomaz Iskra Dragan Vidic Primož Jakopin Janez Kenič Andrej Vibtelič i Dean Mozalič Dva programe su takozvana igra a jedna od njih KONTRABANT (aventura krijumčarenja) ima sve obeležja domaćeg folklora Cilj ova igra ja nabaviti kućni računar koji na želiosi još ovek mora da se krijumčari Kompjuterska igra KONTRABANT ja u suštini putovanje po Jugoslaviji i zato ja kažu dosta teška Da bi sa nagradio trud najboljih i najsirajnijih igrače obezbeđeno je nekoliko primamljivih nagrada medu njima i INTERFACE 1 ZX Specirum Tačna propozicije će biti objevljena na kompjuterskim siranicama slovenačkih listova

Možda čemo ipak imati sraće kažu mladi iz Slovenije Sreća da do kućnih računara ne moramo više da dolazimo – kršenjem zekona Neima Savez socijalisiičke omladina Slovenije (ZSMS) uputio je saveznom i republičkom izvršnom veću predlog da se dozvoli ovoz kućnih i džepnih računara Ako želimo da se oslobodimo kompjuterske nepismanosli za šta se zauzima i ZSM Slovenije potrebno je da svi mladi od osnovna škola pa dalje dobiju osnovna računarska znanja u vaspitnoobrazovnom procesu U tom procesu trabalo bi da učestvuju zavodi ze školstvo komiteli za obrazovanja i vesptanja i druge obrazovne insiliucija

Jugoslavija je jedna od ratkih zamalja u kojima je uvoz kućnih i džapnih računara faktički onemogućen Kada bi se lični računari svrsiali u predmala široka potrošnja bio bi omogućen uvoz i što je još važnije njihove konsignaciona prodaje Ako tome dodamo i razvoj domaće proizvodnja ličnih računara (koji doduša na mogu da konkurišu stranim računatima ni po cani ni po snazi) to bi već bio ogroman korak napred Savezni sekretarijat za spoljnu trgovinu obznanio ja da je u toku rad na izmenama i dopunama Odluka o uslovima pod kojima fizička lica mogu uvozili i primati određene predmeta iz inostranstva Da li će sa nešto promaniti u tratmenu kućnih računara?

U iskri obačavaju da ća sa već ove jesani na tržišiu pojaviti prvi kućni računari HR–84 domaće proizvodnje — prvih 500 kompjutera iz serije od 2 500 komada Do tada bi Iskra trebalo da okonča i pregovora oko konsignacije Sinclairovih računara

Sasvim mali broj Sinclairovih računara ZX81 iznala ja u maju na izložbi u Mladinska knjige iz Ljubljane Medutim to su računan koji imaju svega 1K RAM e što znači pole kucane stranice naredbi pa zato stoji pod znakom pitanja njegova upotrabljivost za osnovna i srednje škole Bilo kako bilo možda su to ipak prve sveila iačka na horizontu računarska pismanosti kod nas Ipak jedno je sasvim sigurno mladi imaju i dovoljno anargije i dovoljno znanja da iznasu kompjutersku revoluciju ali na želost za sada manje-više samo kao samonikli entuzijasti

I baš zbog toga možamo da smatramo prvu slovenačku kasetu sa prvim domaćim programima za značajan događaj za hrabst i pohvalan čin uloliko pra što je softverska redakcije Radija-študenti mlada — oformljena tek ove godine

Tiraž kasete je 2.000 primaraka cena 700 dinara (što ja malo ako imamo u vidu da se u inostranstvu za jedan program plaća od 5—7 funti) Najvažnije u svemu ja da oni za jesen pripremaju kao što su obavestili na konferenciji za štampu novu kasetu sa područje obrazovanja u saobraćaju Dakle prva slovenačka lasta — nije i poslednja

Kristina Iskra

Sajam u los Andelesu je osnovan još davne 1976 Njegov osnivač ja Džim Voran (Jim Warran) — bivši hipik urednik nekoliko kompjuterskih časopisa nastavnik matamatika i na kraju doktor kompjuterskih nauka sa čuvanog Stanford univerziteta blizu San Franciska On je pošto je dobro poznavao situaciju u Silicijumskoj dolini (južno od San Franciska nakad čuvana po šljivicima i suvim šljivama a danas cantar svatske moderne tehnologija) gde sa 1976 puno toga dogadalo u vezi sa kućnim računarima odlučio da organizuja skromnu izložbu All već prve godine interes je pramašio sva očakivanja i mala izložba je prarasla u sajam.

Džim ja uvak bio široke ruka I nije odbijao njednog izlagača — uvak bi se i za najmanju firmu ili pojedinca našao neki kutak Prča sa tako da osnivač APPLE-a Stiv Džob (Steve Job) nija imao nekoliko stotina dolara da 1977 plati za izložbeni prostor ali Džim ga nija odbio, već je pristao da čaka na novac Tako ja Apple-a predstavljen javnosti Znamo šta sa kasnije dogodilo sa ovom kompanijom No Džim je prošla godine prodao sajam jer kako kaže nema više onog zadovoljstva kao prvih dana

Posetilaca sve više, ali nema varnica

Kako je ove godine bilo u San Franciscu? Na žalost nema više malih izlagača koji iznanađuju velikim idejama — sve sama velika kompanije a medu njima gotovo polovina izdavači časopisa i knjiga Kako piše novinar dnevnog lists San Francisko Kroniki (San Francisko Chronicle) posetilaca sve više ali nema varnica Naravno vodeći IBM i APPLE zauzimali su najviše prostora Appla koji ove godine trošl 15 miliona dolara na promociju Mekintoša (kakva razlika u odnosu na onih nekoliko stotina dolara iz 1976!), Izložio je četir mastra visoki Mak na čijem su se velikom ekranu slikovito pokazivala mogućnost ovog računara Bilo je i nekoliko starih modela Apple II/a i Lise ali na njih niko nije obraćao pažnju

IBM-ov izložbeni prostor bio je najviše posvećen novom PC junioru uz stariji PC kompjuter ali ja posetioca ostavio prilično ravnodušnim! Dobri poznavaoci tržišta tvrde da se on slabo prodaje (vaš dopisnik ja obišao više radnji u Kalifornij i dobio je isti utisak) Od ostalih novosti sa sajma ja to da ja OSBORNE posla bankrotstva ponovo oživeo (njagov osnivač Adam Osborne je smanjen — sada vodi jednu softversku kompaniju) Interesantan je bio i štanc japanskog nacionalnog Mikao udružanja — oni su prikazali najnovije proizvoda japanskih kompanija

Sa knjigama i časopisima nastupilo je mnoštvo izdavača Dosta interesovanja izazvala je i humoristička knjiga Mrzim kompjutare (j hate computars) Svaki računar kao i u Evropi prati po nekoliko časopisa Mekintoš ih sada ima već dva (Mackvord i St Mac) a izdavači najavljuju još pet novih

I pored visoka cena ulaznica (12 dolara) sajam je posetilo preko 40 000 antuzijasta Na američkim sajmovima sa uzgred upo redo održavaju i besplatni seminari kojima moze prisustvovati svaki posetilac Na njima govora najpoznatija imena iz ovog područja i *moža* se naučiti dosta toga

na zapadu ništa novo

Računari u sve

U proleća svake godine na zapadnoj obali Amerike održava se nekoliko sajmova za kućna računare na kojima se najbolje može videti koji vetrovi duvaju u ličnoj informatici. Vaš dopisnik već sedam godina napravi bar jedno putovanje preko Atlantika da vidi nove proizvode i trendova. Tako je bilo i ove godine. Poželi smo najatraktivniji sajam te vrste, IX West Coast Computer Faira u San Franciscu, i sajam Comdex (zatvorenog tipa) u Los Anđelesu. Dok je prethodnih godina bilo iznenađenja i uzbuđenja na pretek, prošle, a pogotovo ove godine, ovi sajmovi su počeli da gube svoj tradicionalni šarm. Od kako su poslednjih godina velike multinacionalne kompanije zavladale ovom industrijom, nema više genijalaca koji će u svojoj garaži napraviti novi računar ili u studentskoj sobici sastaviti softverski program koji će se prodati u 100.000 primeraka. Da li starimo ili smo previše romantični?



Na visokoj nozi: Tridesetdvobitni Macintosh nova nada kompanije Apple i čitave američke lične informatike košta svega oko 2.500 dolara

Računari, telegrafi i . .

Sajam u Los Anđelesu je drugačijeg karaktera — trebalo bi da bude samo za ljude iz kompjuterske industrije. Međutim, ko god je bio raspoložen da plati 25 dolara, dobijao je bedž sa natpisom „gost izlagača" koji mu je otvarao vrata u dvoranu. Uz oko 350 manjih izlagača, bili su prisutni i svi računarski džinovi: IBM, APPLE, RADIO SHACK, DIGITAL. Uz dva britanske (Acorn i Memotech), bilo je takođe i nekoliko japanskih kompanija. Najviše prostora zauzela je sa svojim računarom džinovska kompanija ATT (American Telephone and Telegraph) koja ima skoro monopol u Americi na telefonsku i telegrafsku mrežu (za razliku od mnogih zemalja u svetu, ovde je to u privatnim rukama — samo je poštanska služba u državnim).

Pojava ove kompanije na tržištu računara predstavlja prvorazredno iznenađenje. Možda će od toga zaboleti glava mnoge proizvođače — ne toliko zbog samog računara koliko zbog velikih finansijskih sredstava koja namenjuju da ulože u publicitet i propagandu. I pored svih pokušaja japanskoj industriji još uvek ne polazi za rukom da zauzme bole značajniju ulogu na tržištu. Sada su uveli novu taktiku — neki njihovi računari se prodaju pod imenima RADIO SHACK COMPAQ jer ih oni proizvode jeftinije od Amerikanaca. Na tržištu periferijske opreme, posebno štampača i disk jedinica, Japanci vode glavnu reč. Posle poburne koju je ovde doživeo Acorn BBC (ali to je prošlo) najpoznatija britanska kompanija na postiže neki veći uspeh. Acorn se prodaje školskim institucijama, ali dosta slabo jer je sa cenom od oko 1 000 dolara previše skup.

Ove godine dakle, dosta razočarenja. Sledeće će možda, biti bolje.

Anđelko Zgorelec

Evolucija mozga od dinosaura do kompjutera

u susret superinteligenciji

Zamislite tiranosaura reksa najmoćnijeg među dinosaurima: kako se osvrće oko sebe u bujnoj prašumi jednog davnog sveta. Visok je dva sprata; bedra mu se nadimaju od snage; u njegovim razjapljenim čeljustima bleska nekoliko redova zuba oštrih kao brijač: on je jedna moćna mašina za razaranje. Ali u njegovoj ogromnoj lobanji smeštan je majušan mozak. On je glupa životinja.

Dok se mrak spušta, jedno malo dlekavo stvorenje izlazi iz svog skrovišta gonjeno ljutom glađu. I kreće oprezno u okolno šiprašje po svoj obrok crva i insekata. Tog trena ogromne telesina tiranosaure zamrečuje nebo: njene senka pada na malu životinju i ova žurno beži ražeći spas.

To stvorenje je jedan od prvih sisare na Zemlji. Ono nije impozantna borbena mašina, ali je intelektualni džin svoga vremena, ono je najinteligentniji oblik života koji se razvio na planeti do tog perioda. Srazmerno težini svog tela ono je 5 puta pametnije od tiranosaura a 20 puta pametnije od bilojednih dinosaura kojima sa tiranosaur hrani.

Zašto je taj reni sisar bio toliko pametniji od dinosaura? Niko nema pouzdan odgovor, ali verovatni razlog je taj što je on tada imao malu šansu na uspeh a pritisci pod kojima je živeo nametali su potrebu velike inteligencije u borbi za opstanak. Malo stvorenje mora da je bitisalo u stanju nepreklidne strepnje krijući sa preko dena i tregajući noću za hrenom pod mučnim i nesigurnim okolnostima. Besponočno moralo je da živi od svoje pameti.

Made su se siseri i dinosauri medusobno veoma razlikovali, oni su bili rodaci jer su poticali od iste loze drevnih reptila. Dravni reptili sa svoje strane razvili su se iz emfibija, a emfibije su potekle od riba. Svi ti oblici života — ribe, amfibije i reptili — bili su relativno neinteligentni, a i njihovi potomci ostali su takvi sve do danas. Ali sisari su bili drukčiji: priča o evoluciji inteligencije u stveri je *njihova* priča. Pojava sisara označile je prvi veliki korak u evoluciji mozga.

Gospodari planete

U početku sisari — ili tačnije rečeno reptili nalik na sisare, dakle jeden prelazni oblik — bili su sviepi mnogobrojni i strahovito uspešni u stvari pre 250 miliona godina oni su bili dominanten oblik života na kopnu. Ali kesnije, iz nekog rezloga opali su u broju i veličini i potonuli u tamu dok su ih dinosauri nasledili kao gospodari planete.

Sisari su ostali potčinjeni vladajućim dinosaurima više od 100 miliona godina, promenivši se vrlo malo tokom tog dugog intervala. A onda, otprilike 70 miliona godina unazad, klima na Zemlji počela je nagio da se menja, i dinosauri, ne više u harmoniji sa svojom životnom sredinom, postepeno su iščezli. U roku od svega nekoliko miliona godine potpuno su nestali. Sisari, sada odjednom nsaputani, uselili su se u životna prostore koje su upraznili mrtvi dinosauri i rascveteli se u čudsano mnoštvo raznovrsnih oblika. Neki su se opredelili za vazduh i postali leteći drugi su sa vratili moru i postell kitovi i delfini, treći su ostali ne šumskom tlu i jedna grupa se uspentrele na drveće de bi postale preteče majmuna i čovekolikog majmuna.

Poput njihovih srodnika koji su istraživali i osvajali druge životna prostore ovim životinjama nastanjenim na drveću bile su potrebne specijalne telesne odlike koje će ih opremiti ze taj novi život: ruke ze dohvatenje grena na primer i binokuleren vid za procenjivanje rezdaljine od grane do grene. Tokom vremana te odlike su evoluirale. Ali stanovnik drveta morao je da ima veći i bolji mozak. Kad je skakao s jedne grane na drugu bio mu je potreban brz, precizen kompjuter u glevi za kombinovanje faktora razdaljine, brzine vetre, pokrete grane i revnoteže tele. A izračunavanje je moralo de bude obavljeno u magnovenju, jer bi eveka pogreška bile plaćena smrću ili teškom povredom. Tom prirodnom kompjuteru — mozgu — bile su potrebna pretinjena mentelna kola za obavljanje neophodnih proračuna: a bile mu je potrebna i velika banke memorije za skladištenje rezultata bivših iskustava u vazdušnim manevrima.

Drevni sisari koji su živeli na šumskom tlu posedovali su jednostevniji mozak bez tih izvenrednih kvaliteta. Ali još jednom zakon opstenka najsposobnijih stuplo je u dejstvo. Zahvaljujući malim varijacijama od jedinke do jedinke koje se dešavaju u svekoj populeciji, neka životinje su posedovale nešto bolji mozak od drugih te životinje su imale veće šanse na opstanak, prenoseći svoju superiorniju moć mozga na potomstvo, dok su one koje su bile u manjoj meri obdarene tim mentalnim odtikama pokazivale sklonost da nestanu u ranom dobu, i njihovi geni su iščezli iz populecije.

Dejstvo selekcije

Pod dejstvom prirodnog odabiranje mozak stanovnika drveta poboljšao se i u pogledu kvaliteta i u pogledu veličine, istovremano njegova manuelna spretnost i oštrina vide nastavili su da se razvijeju. Nakon 40 miliona godina neprekidnog poboljšavanja tela i mozga potomci prvog stanovnika drveta postali su nova vrsta stvorenja. Ta nove stvorenja bile su majmun — i njegov rodak, pračovekoliki majmun.

Majmun se nije mnogo promenio od tog vremana do danas; njegova je priča bila završena. Ali evolucija čovekolikog majmuna se nastavila. Postao je veći i teži, i sišao je sa drveta da bi otpočeo novo bitisanje na šumskom tlu: vrativši se prebivalištu koje su njegovi majušni preci napustili 40 miliona godina pre toga.

Sada je bila postavljena pozornica za još bržu evoluciju inteligencije.

Pre otprilike 15 miliona godina jedna napredna grupe čovekolikih majmuna — predeke šimpanze i gorile — nastanjivela je šume istočne Atrike. Još jednom promena klima na planeti skrenula je tok evolucije. Nastupile je era hladnijeg i suvljeg vremene uslovivši pojavu otvorenih komada zemljište za napasanje. Dok je suvo vieme nastavljelo da traje, pašnjaci su se sve više širili, i teren se transformisao u poznatu afričku savanu.

Životinja slična čoveku

Savana je imala privlečnu draž nepoznatog za čovekolike majmune. Postepeno, najsmeliji i najradoznaliji među njima nepustili su bezbednost drveća da bi istražili tu čudnu novu sredinu. Ne znamo šte se iza toga desilo jer u fosilnim nalazima tada nastupa jedna praznina od nekoliko miliona godine. Znamo samo da se šumskim čovekolikim majmunima otvoreno zemljište dopalo i da su ostali tamo jer kade se fosilni zapis ponovo pojavljuje on pokazuje da su se njihovi potomci uveliko odomaćili u savani. Ali potomci nisu više čovekoliki majmuni: oni su jedna nova životinja koja hode uspravno korakom nalik čovekovom. A ta nova životinje ima superioran mozak u odnosu na telesnu veličinu: mozak je dvaput veći od mozga čovekolikog majmune.

Prema fosilnim zapisima, ta dvonoga inteligentna zivotinja pojavile se u Africi pre tri do četiri miliona godina. Njeno ime je australopitek afarenzis. Bio je to sposoban lovac koji je jurio sa ostalim mesoždernima svog vremena nadmećući se sa lavom i dzinovskom hijenom. Medutim australopitek je bio slebašan: nejake grade, težak mozda samo oko trideset pet kilograma i bez oštrih očnjaka ili drugog prirodnog oruzja. Ali njegove fizičke slabosti bile su kompenzirane snagom njegovog intelekta.

Koje su okolnosti stvorile tu neobičnu moć mozga australopiteka? Odgovor nije jasan, ali proučavaoci ljudske evolucije uglavnom se slažu da je uspravno dižanje verovatno bilo ključni faktor. Sa rukama odnedevno oslobodenim za nošenje hrane i oružja i za izradu i korišćenje oruda, australopitek je imao potencijal za stvaranje kulture. Ali kako je taj potencijal bio realizovan? Kako su te telesne odlike dovele do poraste mozga?

Ako je prošloet neki putokaz ze budućnost, onde možemo očekiveti nastenak jednog višeg oblika inteligencije. Kako će izgledeti tej prefinjeni mozek? U čemu će se razlikovati od našeg? Da li elektronski računari, pa i ovi kućni, predstavljaju semo primitivni zametek buduće superpameti ili se radi o strenputici na putu za budućnost? O toma piša Robert Džestrou (Jastrow), američki naučnik, publicista i asejista.



Teorije o prirodnom odabiranju nude odgovor. Prvi izradivači oružje u loj grupi dvonogih čovekolikih majmuna mora de au se razlikovali medusobno po stepenu dovitljivosti i ingenioznosti koje su upražnjavali u svom svekodnevnom životu. Oni koji su posedovali te osobine u najvećem stepenu bili su istinski juneci dana. oni su imali najviše izgleda da načine elikesna oružje za ubijenje divljači i najboljo strategiju ze lov. Oni su imali uspeha oni i njihovi potomci su napredoveli njihovi geni au se sve više ukorenjivali u populaciji odlike dovitljivosti i ingenioznosti intenzivirale su se medu dvonogim čovekolikim majmunima.

Dovtljivost i ingenioznost — lo su dve lica inteligencije a njihovo sedište je u mozgu. Ako se ovakva odlike intenziviraju mozak mora da poraste. Ne tej nečin je uspravno držanje koje je dovelo do izrade i korišćenja oruda doprinelo ubrzenjem porastu mozga u populeciji dvonogih čovekolikih majmuna.

Automatske poruke

Mozak koji je rezultireo nije bio velik u epsolutnom smislu. imao je tek veličinu pesnice i težio jedva 450 grama: e druge strane medutim ni telo koje je kontrolisao nije bilo veliko. U procenjivanju inteligencije neka životinje po veličini njenog mozge važno je izračuneti proporciju veličine ili težina mozga u odnosu ne telesnu težinu. zato što svaka životinje koristi jedan deo svog mozga ze nesvesnu kontrolu tele. Tej segment mozga služi kao teletonska razvodna tebla koje prima signale od čula i se svoje strane odašilje automatske poruke. Beži od opasnosti! Približi se hrani! I teko delje. Samo višak sive materije koji preostaje posle tih automatskih funkcija, stoji ne raspolegenju za postupke koji mogu da dodu pod kontrolu inteligencije kao što su pamćenje plenirenje i fleksibilno, neprogramirano reagovenje ne nove izazove. Australopiteku sa melim telom. bio je potreban relativno mali broj moždanih ćelija za telefonsku razmenu u njegovoj glevi pa je zato imao ne raspolaganju veću količinu sive materije ze pamćenje plenirenje i apstraktno mišljenje.

Australopitek je stekeo tu moć mozge pre dva ili tri miliona godine nekoliko miliona godine nakon što su se njegovi drevni preci prvi put preselili u savanu. Izmenio se donekle tokom nekoliko sledećih miliona godina. evoluirajući uspešno u životinju poznatu pod imenom australopitek afrikanus a zatim u telom krupnijeg australopiteka robustusa. Posle toga se promenio veoma malo. keko u pogledu mozga teko i u pogledu tela. Paput mnogih drugih oblika živote u prošlosti australopitek je dospeo u čorsokak. Pre sedamato pedeset hiljeda godina potpuno je izumro.

Pre nego što će iščeznuti australopitek je iznedrio drugu inteligentnu životinju afričke savane. Novo stvorenje je nazveno homo erektus — čovek koji stoji uspravno. Njegovo držanje bilo je uspravno kao u australopiteke eli plemenitije njegovo telo i crte lica bili au više čovekoliki a menje majmunoliki: e njegov mozak bio je veći.

U početku razlike u veličini mozga nije bila osetne. ali inteligencije australopiteka ostela je gotovo nepromenjena. dok je inteligencija homo erektusa i dalje rasla. Najnovije etapa u evoluciji inteligencije na Zemlji lek što nije bila olpočela.

Neumitni trend evolucije

Sade nas je priča dovele do jednog vremena pre otprilike milion godine kada je australopitek upravo trebalo da iščezne iz savane. Homo erektus već je lovio sa zapanjujućom veštinom. nijedne divljač nije bila suviše velika za njegovo junaštvo. e njegova tehnologija kamenog oruda bila je impresivne. Odjednom. rast mozga se ubrzao. Niko ne zna tačno zašto. smatra se da oruda. jezik i pritisci ledenog doba spadaju medu glavne razloge. eli me šta da je bilo uzrok tosilni nalazi demonstriraju rezultate.

Blizu 450 grama sive materije bilo je pridodeto ljudskoj lobanji za manje od milion godina. Veći deo bio je u prednjem delu mozga koji je center najviših domena epstrakcije i kreativnosti. Dok su se dodetni neuroni gomilali u lobanji renog čoveka. njegovo čelo se ispupčavelo lice je izgubilo svoj zverski izrez i on je počeo da poprima izgled intelektualnog bića. Homo sapiens — čovek obdaren razumom — stigao je na planetu.

Danas čovek predstavlja vrhunsku tvorevinu prirode na Zemlji. Kakva budućnost predstoji toj izuzetnoj životinji? Možda će i on iščeznuti kao što je australopitek izumro pre njega. devedeset odsto svih oblika života koji su postojali na Zemlji vremenom su izumrli. Ti će možda živeti neizmenjen sve do u deleku budućnost — jedan živi fosil poput školjke kamenice. Ta sudbina možda se već nadnela nad nas. jer ljudsko telo se izmenilo veoma malo tokom poslednjih milion godina. a ljudski mozak se nije izmenio bar ne u ukupnoj veličini tokom protekłih 100 000 godina. Organizacija mozga možda se poboljšala za to vreme. ali ograničene su količine informacija i uzajamnih poruka koja mogu da budu strpane u lobanju. Iskrsnea veličine činjenica da se mozak više ne širi posle milion godina eksplozivnog rasta sugeriše da je priča o ljudskoj evoluciji možda već završena.

To na znači da je završena i evolucija inteligencije. Naime razumno je pretpostaviti da ljudska biće na predstavljaju poslednju reč u evoluciji na Zemlji. već samo izvornu lozu iz koje će se razviti jedan novi i viši oblik života. da bi prevazišao naša dostignuća kao što smo i mi pravazišli dostignuća australopiteka. Istorija života potvrđuje taj zaključak. jer pokazuje naumoljiv trend ka većoj inteligenciji kod viših oblika životinja. Očigledno. među svim odlikama nekog živog organizma. nijedna nema veću vrednost za opstanak od fleksibilnog inovacijskog reagovanja na izmenjene okolnosti. reagovanja koje nazivamo inteligencijom. Izgleda neverovatno da će taj trend u evoluciji biti iznenada okončan na jednom određenom nivou mentalnog dostignuća koji nazivamo ljudskim. Ako je prošlost naki putokaz za budućnost čovečanstva ja izgleda. sudeno da dobije jednog još inteligentnijeg naslednika.

Intelektualac

Kakav će izgled imati naš naslednik? Sudeći po skorašnjoj čovekovoj istorji. novi oblik života ličiće na stari. ali će imati još veći mozak. Prema tom predvidanju sledeća vrste inteligentnog života na Zemlji biće intelektualac stvorenje slično nama ali sa većom lobanjom i slabijim mišićima.

Međutim izvesni trendovi u modernoj tehnologiji sugerišu sasvim drukčije videnje budućnosti. U dejstvu su moćne sile evolucije — preveshodno kulturne a ne biološke — koje mogu da dovedu do jednog egzotičnijeg oblika inteligentnog živote — oblika koji će se razviti iz čoveka. ali pre kao dete njegovog mozga nego kao deta njegovih slabina.

Prema tom videnju. novi oblik života se danas već stvara u laboratoriji kompjuterskog neučinka. To je jeden veštački zivot načinjen od silicijumskih čipova a ne iz neurona. I ipak on misli pamti uči se na iskustvu. i reaguje na nadražaje. Njegovo razmišljenje je još uvek priprosto. nije mnogo kreativen. ali evoluira zapanjujućom brzinom.

Ideja. Izgleda apsurdne. kako može bogatstvo ljudske misli da bude uporedivano sa mehaničkim razmišljanjem jednog kompjutera? Pa ipak. Iznenađujuća stvar kod kompjuterâ nije u tome što oni misle manje dobro od čoveka. nego što uopšte misle. Tačno je da su elektronski mozgovi današnjice primitivniji u poređenju sa ljudskim mozgom. u stvari. oni nam daju malo razloga da im se divimo. sem zbog svoje ogromne memorije i izvesnih matematičkih veština. Međutim najnoviji modeli mogu biti uprogramirani lako da prate neku raspravu. postavljaju drčna pitanja. čak da pišu dopadljivu poeziju i muziku. Isto tako oni mogu da obavljaju ponešto neobavezne konverzacije toliko ubedljivo da njegovi ljudski partneri nisu ni svesni da razgovaraju sa mašinom.

Silicijumski mozak

To su dopadljivi kvalitati kompjutera. on imitira život kao neki elektronski majmun. Dok kompjuter postaje kompleksniji. i imitacija postaje bolja. Na kraju crta koja razdvaja original od kopije postaje zamagljena. Kroz sledećih petnaestak godina — oko 1995 prema tekućim trandovima — mi ćemo videti kako se silicijumski mozak javlja kao jedan novi oblik života koji može da se takmiči sa čovekom.

Pre nego što odbacimo tu viziju razmotrimo neka skorašnja dostignuće u kompjuterskoj industriji. Prvu istinski modernu generaciju kompjutera reprezentovala je mašina IBM 360 koja je puštena na tržište 1964 godine. Jedna mašina tog tipa obavljala je posao za Godardov institut za proučavanje svemira tokom poslednjih 15 godina. Koliko bi velika bila takva mašina ako bi je trebalo proširiti toliko da odgovara kapacitetu ljudskog mozga?

Mozak je težak približno 1 400 grama troši elektricitet snage otprilike 20 vate i zauzima zapremlnu tridesetog dela kubnog metra. Unutar tog malog prostore on sadrži memoriju od 10 milijardi informacionih jedinica. Jedan kompjuter IBM 360 tog isiog kapaciteta trošio bi električnu energiju od milijardu vata. ispunjavao bi čitav Svetski trgovački centar. i koštao bi mnogo milijardi dolara. Bila bi to nezgrapna imitacija mozga.

Ali najnovije mašine na tržištu 1980 samo 16 godina kasnije. pričaju drukčiju priču. Jedna od tih novih mašina proširena toliko da odgovera kapacitetu mozga mogla bi se smestiti u prostorije neke kancelarije. i trošila bi samo 100 000 vata elektriciteta. Bila bi još uvek nezgrapna. ali gotovo vredna izgradnje.

Portabl kompjuter

Dalja poboljšanja. koja se sada odvijaju u laboratorijama za kompjutere. biće spremna za tržište oko 1990 godine. U to vreme moći će se izgraditi kompjuter čovečjeg kapaciteta koji bi mogao da se smesti u koter i da operiše pri utrošku električne energije od 1 000 vata.

Oko 1995 krivulje rasta silicijumskog mozge mogla bi da prede crtu ljudskih sposobnosti — 10 milijardi činjenica u jednoj aktentašni. sa utroškom od svega 20 vata električne energije.

Naravno. silicijumskom mozgu koji odgovara ljudskom mozgu samo po smeštajnom kapacitetu i moći grubog resudivanja nedostajaće kreativna inspiracija i intuicije koje karakterišu više domena ljudske misli. Poput nekog novopečenog doktora filozofije. on će imati ogromne mentalne moći ali ne i mudrost. Pa ipak. i tu kvalitativnu superiornost mozga nad kompjuterom sve više podrivaju izvesna novija razvojna kretanja. Da bi sa shvatio značaj tih kretanja razmotrimo za časak arhitekturu ljudskog mozga. Svaka moždana ćelija je direktno povezana sa mnogim drugim moždanim ćelijama. u nakim slučajevima čak sa 100 000. Kao rezultat toga. kad god pošaljemo neki svesni impuls dola u dobine naše memorije da prizovemo neku odredanu informaciju. ćelije u kojoj je ta informacija pohranjena komuniciraju na podsvesnom nivou sa hiljadama drugih ćelija. I jedan talas mentalne aktivnosti prostruji kroz čitav mozak. Plodovi te podsvesne aktivnosti su intuitivna spoznaja. bleak percepcije i nadahnuća genija. Što je sve omogućeno zahvaljujući bezbrojnim vezama među ćelijama ljudskog mozga. Te veze među ćelijama objašnjavaju mnogo šta od neobične moći mozga.

Arhitektura kompjutera je sasvim drukčija. Kompjuterske memorije su kao jedna garnitura otvora na golubarniku poređanih uza zid. bez povezanosti jednog otvora sa drugim. Informacija može da bude stavljena u golubiji otvor. Ili izvadena iz njega. ali golubiji otvori ne komuniciraju međusobno niti razmišljaju. Razmišljanja se obavlja na drugom mestu.

Integrisana kola

Najnoviji razvoj koji menja arhitekturu kompjutera baziran je na izumu koji nazivamo čipovima. U svom najnovijem obliku čipovi nose naziv integrisana kola. To su tanka četvrtaste pločice silicijuma duge otprilike šezdeset milimetra po jednoj strani. koje su povezane medusobno provodnicima u velikom broju. one formiraju mislaće i memorijske sekcije elektronskog mozga. U starim danima. oko 1960 ili 1970 svaki silicijumski čip mogao je da drži samo mali broj tranzistora i elektronskih elemenata. Rezultat toga bio je da je jedan čip mogao ili da misli. ili da pamti. ali ne oboje. Danas. međutim. vodeći proizvodači integrisanih kola u Sjedinjenim Državama i Japanu naučili su kako da natrpaju 100 000 tranzistora i elektronskih elemenata na jedan jedini čip. Sada jedan čip može i da misli i da pamti. Ako je svaki čip povezan sa mnogim drugim čipovima kompjuter postaje pravi silicijumski mozak poput ljudskog mozga. on funkcioniše kao jedna jedinica. obavlja misaone procese u lalasima unutrašnje mentalne aktivnosti na podsvesnom nivou. Ovde se već približavamo onoj vrsti asocijativnog mišljenja koje predstavlja tako moćan elemenal u ljudskom rasudivanju.

To su izgledi koje možemo predvideti za poslednju deceniju našeg veka. U to vreme silicijumski mozgovi će u saradnji sa našim najboljim umovima raditi na svim ozbiljnim problemima dana. u jednoj nepobedivoj kombinaciji moći sirovog rasudivanja i ljudske intuicije. Šta će se zatim desiti? Šle u sledecem stolecu? Iâte u stoleću iza toga? Ne mogu se predvideti granice rastuće krivulje silicijumske inteligencije. Jedan od vodećih ljudi u kompjuterskom istraživanju profesor Marvin Minski sa MIT-a veruje da će se moći ostvariti masine sa opštom inteligencijom jednog prosečnog ljudskog bica — mašina će početi da obrazuje samu sebe — kroz nekoliko meseci će biti na nivou genija — a nekoliko meseci posle toga njena će moć biti neizračunljiva. A onda kaže Minski. Ako nam sreća bude naklonjena. one će mozda odlučiti da nas drže kao kućne ljubimce.

10/ evolucija mozga od dinosaura do kompjutera

Računari i mladi

era tehnoloških genija

Mikroelektronske revolucije, koja je, sa prilično zakašnjenje, najzad zapljusnule i naše obale i za sobom povukle ne hiljeda mališane, zatekla je roditelja i pedegoge potpuno nespremna. Šta se to najednom dešava sa mladima? Da li su se oni napreèac okranuli „korisnim" stvarima ili i njihov novi hobi ima svojih skrivanih zemki i opesnosti? Koja je stverno masio računara u obrazovanju, e koje u životu? Šte očekuje ona koji sa protesionaino opredele ze računarstvo, a šta ona koji ostanu „kompjuterski nepismeni"? Ove stranice, koja prenosimo iz knjiga „Računeri i veše dete" engleskog sociologa Reja Hemonda (Ray Hammond), treba da pomognu (zbunjenim) roditelijime de bolja shvata odnos dece prema računarima i previlnije usmare njihov rezvoj.



Na muškoj teritoriji. Interesovanje za računare je etrogo podvojeno na polove — ne zato što eu devojčice za njih „glupe" a dečaci „pametni" nego isključivo iz sociјalnih razloga

U mnogim delovima sveta mladi se obrazuju u sistemima koji su u haotičnom stanju Masovni priliv računara uz povećanu verovatnoću da se posle završenog školovanja neće moći doći do zaposlenje lišio je većinu obrazovnih sistema svake koherentne politika Svedoci smo od 1960 godine luda trka da sa mladi donekle kompjuterski opismene pra nego završe školu sa širanjem recesija u svetu pretnja nezaposlenosti ja tu tendenciju samo pospešila Većina roditelja smatra da njihovo dete treba da razume kompjutar ali ni dve osoba. Izgleda ne mogu da se slože kako da se to ostvari ili u čemu bi se to sastojalo

Greška u koracima

Nezaposelenost je posebno oštra medu onima koji završavaju školu U nestojanju da mladima otvori veće mogućnosti zaposlanja većina srednjih škola pruža sada neki oblik kompjuterskog obrazovanja To sa najčešće radi tako da sa daci predaja programiranja u okviru nekog kursa kompjuterskih studija Na ovim kursevima najčešće se predaja bejzik ali mnogi nastavnici već razmišljeju i o alternativnim kompjuterskim jezicima kao što su komal i algol

Deca mlada od 12 godina u stvari nisu u stanju da rada ni sa jednim od ovih apstraktnih algoritamskih jezika i u mnogim srednjim školama obrazovanje u računarima ne počnja pra nego što dete ude u treću godinu (osmi ili deveti razred)

Postoji mišljenje da je pogrešno uvoditi mlade u svet računara tako što će im se predavati programiranje To ja smatra se kao da se onoma ko treba da nauči da vozi avion govori o tome kako je on konstruisan Odvajajući računarstvo u posaban studij i tražeći od učenika da dolazi u računarsku sobu (često locirnu u matematičkom kabinetu) škole odvajaju računarstvo od drugih običnih predmeta i ističu njegovu matematičku ili naučnu osnovu Ta se situacija često uporno održava jer te nastavnik matematike ili egzaktnih nauka prvi uveo računarstvo u školu pa nestoji da zadrži ličnu vlast nad svojom teritorijom

U nekim školama računarski predmeti se slabo probijaju u osnovni nastevni program pa se računarstvo još uvek predaja kao grana nauka Po jednom interesantnom tumačenju računarska nauka je se drugim predmetima kao što su ekologija i fizičko obrezovanje izbačena iz nastavnih programa jer su ambiciozni nastavnici pažnju usmerili na predmete koji im donose više prihoda i daju više vlasti u njihovim institucijama Postojeći nastavni programi u srednjim školama ispitno su usmereni sposobnost kako učenika tako i nastavnika odreduja se samo uspehom ne ispitima Ugled unapredenja i novac idu nastavnicima koji predaju teorijska predmete koji se najlakše mogu uključiti u ispitni format Izgleda da ima dosta dokaza u prilog ovog mišljenja kao neposredna posledica ovog i drugih pritiska u mnogim srednjim školama računar ostaje zaključan u matematičkom ili naučnom kabinetu

Rezervat za dečake

Ovo odvajanje računara ima i drugu tešku posledicu Medu dacom izmedu 11 i 18 godina računari i računarstvo smatraju se prvenstveno „muškim" predmetom To ja delimično stoga što se računar često čuva u muškim" odeljenjima — naučnim ili matematičkim — pa je njegovo korišćenje još uvek u vlasti muškaraca, a delom zbog načina na koji sa računar koristi u nastavi

Razlika u intelektualnom razvoju polova dugo su predmat istraživanja i suprotnih gledišta, postoje neki površni dokazi da su u toku mladelačkih godina dečaci u matematici uspešniji od devojčica Ovakvi nalazi sa moraju primati a rezervom jer mogu biti

posledica načina na koji se devojčicama predaje matematika. Uticajne studije britanske vlade Socijalni trendovi koje se objavljuje svake godine, pokazuje da su devojčice u celini uzev, uspešnije od dečaka: da 32% devojčica koje završe srednju školu odlazi na dalje školovanje, prema 22% kod dečaka. U najboljim grupama međutim, dečaka je nešto više od devojčica.

Ono što je međutim jasno, jeste da računari pripadaju dečacima — dar u ovom trenutku. Zahtevi tinejdžere (od kojih su mnogi ludi za vasionskim igrama) pristižeju mnoge porodice da nadave računar. Proizvođači su svesni ogromna snage tog tržište i koriste je, podstičući postojeće trendove. U ranom mladićačkom dobu seksualna uloge su polarizovane. Kada dečaci i devojčice počnu da otkrivaju svoju seksualnost oni sa povlače svako u svoj ugao kako bi razmotrili situaciju. Na tom tržištu oglasi i izložbe kućnih računara uglavnom počivaju na igrama koje sa nepogrešivo odnose na divlje retove u kosmosu i herojska dela — ava muška po pristupu. Proizvođači se na usuduju da tinejdžerima njihovo interesovanje za računare prikažu kao delimjeto — radi se o grupi nedovoljno odraslo da spozna i kontroliše svoj divlji seperetistički šovinizam. Tako se računar svrsteva u kategoriju kartinga i tutbala, e devojčice koje nemaju sreće da u njemu otkriju ono što ih interesuje celu stvar tako i klasifikuju.

Kućni računari se masovno kupuju na tinejdžerskom tržištu. Pre dve decenije tako je bilo sa električnom gitarom. I kao što su se šezdesetih godine roditelji nadali da će njihovo dete napredovati i stići do prevog instrumenta, tako sa današnji roditelji nadeju da će osvojači kosmosa dovesti do konstruktivnijeg pristupa primeni računara. Verovatno je da će oni u tom svom nadanju imati više sreće nego što su je imali njihovi roditelji.

Akciona igra su izvanredna atrakcije za dečaka (mada ih vole i mnoge devojčice). Previlno usmereni dečaci i devojčice će napredovati ka krastivnijem korišćenju računare. Izgleda da neka deca posteju veoma vezana za pakmena ili njegov ekvivalent eli iako se često koristi reč zavisan (kao u slučaju droge) name dokaze da dolazi do teko ozbiljnog stanja.

Veliki igrač

Veoma se mnogo istražuju motivi koji navode mlede da troše novac i slobodno vreme na akciona igre. Kod pakmena na primer, svaka uništena tačke predstevlje dostignuće trenutni uspeh a ako naša igre nije vrhunska protivnik — mašina — nije u mogućnosti da nam se podsmehuje. Igre deju učesnicima mogućnost da se istaknu i veoma vešti igreći mogu osvojiti zavidan igrečki ugled. Da bi se dalje shvatile privlečnost akcionih igare treda razmotriti šta ona stvarno deju. Jsano je da igreći bivaju potpuno epsorbovani — osećanje koje bi se teško mogto ostvariti u mnogim drugim aktivnostima. O toj epsorbovenosti se govori kao o posebnom duševnom stanju u kome se osoda koncentriše na trenutni zadetek. Iaklučujući sve druge unutrešnje ili spoljne stimulense. Akcija i svest se



Status quo. Devojčice u Engleskoj za znatno rede odlučuju za računarske studije od dečaka i taj odnos se već godinama ne menja

atepaju tako da osoba prosto radi ono što se more uraditi, bez kritičke dvojne perspektive sopstvenih ekcije. Nsatoji se da ciljevi budu jasni, sredatve se koordinirаju sa ciljevima i povratne sprega sa ekcijom je neposredna i jednoznačne. U takvoj situaciji osoda ima jak oaećaj kontrole — ili lične motivisanosti — pe ipak što je paradoksalno uključivanje ega je slabo ili nikakvo. teko da se doživljava osećaj sopstvene transcendencije. ponekad osećaj jedinstve sa okolinom. Izgleda kao da je proticanje vremena poremećeno naki događaji keo da treju nesrezmerno dugo ali časovi uopšte uzev prolaze u minutima.

Najbliže poredenje stvernog života sa doživljejem akciona igre verovatno je borda ili izvršenje nekog izuzetno hrebrog dela. Duh i telo su stopljeni u jedinstveni veličanstveni cilj, lišeni svega drugog. Radost koje se doživljava, zajedničke je simulirenim i stvarnim akcijama. Beskonečno vežbanje i dresiranje vojnika upravljeno je ke postizanju tog atapanje svesti u trenutku kad sa suočl sa neprijateljem. Još nije utvrdeno de li to zemenjuje potrebu mladih za naalničkim delovenjem.

Programiranje na stramputici

Kada su se pojavili 1975 godine kućni računari (i veći deo školskih mikroračunera) mogli su da se uporede sa kristalnim detektorima koje su u ranijim generacijame konstruisali mnogi daci (ali vrlo malo devojeka!). To je bio hobi u kome su teškoća konstruisanja bile sastevni deo privlečnoati. Ali računerski jezici su se usavršavali sa usavršavanjem računare i verovetno je de u vrlo bliskoj budućnosti nećamo moreti razumeti računerski jezik da bismo mogli programirati računare — kao što danas ne moremo biti u atanju da podesimo oscilator da bismo mogli da izaderemo radio -stenicu. Za nekoliko godine računerski jezici kao što su dejzik ejgol fortren i kobol biće zastareli za normalno programiranje i biće samo od akademskog značaje. Sa porastom kapeciteta računara i inteligencije softvera detaljno programiranje će postajati sve manje nužno. Već postoje programi. keo što je. Poslednji koji sami pišu programe za korisnike i negovešta vajući vreme kade nam neća biti potredan nijedan kompjuterski jezik. Iako je to odaćenje u ovom trenutku pomalo optimističko jesno je da su dani peekinge i pokinga prošli. (Mislim de te dve reči iz programerskog žargona sumiraju ave što je najgore u mikroračunerenju. One opisuju postupke programiranja koji su već pedesetih godina smatrani primitivnim eli za koje i današnji pokloničl kriatalnih detektora pokazuju najveće interesovanje.) Ono što će biti potrebno jeste potpuno razumevanje načine na koji redi kompjuterske logika. Bez toga se neće moći koristiti programi keo što je. Poslednji koji samo automatizuju pisanje programa.

Uprkos predstojećoj pojavi računere koji će sami sebe programirati škole nastavljaju da predaju programiranje odvajajući računer od glevnog nastevnog programe. Svi deci treba da uče kako redi računerska logika ali mučenje nad bilo kojim posebnim računarskim jezikom prevazideno je i treba ga što pre napustiti. Samo neznatan broj dece nastaviće školovanje sa ciljem da se obrazuje kao profesionalni računerski programer. kad jednom dodu na univerzitet ili ne neki profesionalni kurs oni će oddaciti svu veštinu u dejziku ili nekom drugom višem jeziku i počeće da uče keko se sa mikroprocesorom direkino saobraća korišćenjem mašinskog koda ili asemblerskog jezika. Temeljno poznavanje računerske logike biča ovim dacima korisnije od učenja nekog posebnog višag programskog jezika.

Fanatici koda

Nekoj deci posebno inteligentnijoj pro gramiranje računara može postati neka

vrsta droge, čak mnogo više nego akcione igre koje su ih, možda, prvo privukle računaru. Često socijalno nedorasli, ovi fanatici koda razvijaju sklonost ka računarskoj inteligenciji hrabreći roditelje koji smatraju da je njihovo inteligentno dete na putu da postane tehnološke zvezda; oni se povlače iz realnog sveta u svet spavaće sobe ispunjen nezgrapnim zastarelim računarskim jezicima. Ta situacija postaje nazdrave kad dete počne da piše programe u potpunosti za sopstveno zadovoljstvo. Privatni svet spavaće sobe postaje svetilište u kome je dete bog a računar pruža detetu i roditeljima opravdanje za potpuno iracionalno i štetno ponašanje.

Nepoznavanje računarske meterije često dovodi do situacije u kojoj se te deca iskreno ali pogrešno smatraju zvezdama. Nastavnici ne mogu da idu u korak sa detetom kada se ono dubije upusti u mogućnosti veštačke inteligencije i potpuno su nesposobni da procene stvarnu vrednost dečijih dostignuća. Često dolazi do velikih zloupotreba programiranja i dete nije u stanju ili nije voljno da prihvati nikakav praktičan savet u vezi sa svojim radom na programiranju. Za trenutak dete impresionira svoje vršnjake roditelje i nastavnike složenim programima sa bleštavim ekranima i zvučnim efektima. U stvari dete je pobeglo iz realnog sveta i navikava se na programiranje u potpunom akademskom vakuumu.

Pod pretpostavkom da ovi dečaci ostanu dovoljno konformisti da bi zadovoljili uslove prijema univerziteti obično moraju da ih reprogramiraju pre nego što ih usmere na studij kompjutera.

Postoji mišljenje da nebavke kućnog računara može poneked da ima još efekt ne učenje mladog čoveka i čak na izgleda za njegovo zapošljavenje. Prema tom mišljenju davanje računara mladime u osetljivom periodu njihovog života u nedi da se tima postavlja osnova za uzbudljivu karijeru može u stvari imati suprotan efekat. Da br na računarima radili keko treba programeri moraju da misle na jeziku koji koriste, keo što je i za tečan francuski ili nemački jezik potrebno misliti na tim jezicima. Večina jezika koji su denas dostupni za kućne računare zastarele je i prefrpana svim i svačim. Ako se mlade osoba duboko uživi u njih biće joj otežano učenje novog jezika.

Jedna mogućnost da se te teškoća izbegne jeste proširiti pristup računarima. Deca pre treba de uče šte oni mogu da urade nego kako da ih programiraju. To znači ked je reč o kući da roditelji treba da obezbede što svestranije korišćenje računara. Što se škole tiče računar treba izvući iz muškog naučnog odeljenja i uključiti ga u ceo nastavni program tako da bude od koristi svim predmetima. On mora postati učilo koje će se normalno koristiti ravnopravno se globuelima i vrdeonkoredermа.

Nastavnici računarskih predmeta retko stvarno poznaju predmet koji predaju. To je javno izjavila jedna britenska komisije koje je našte de su nedevni ispiti na kompjuterskim studijema A-nivoa otkrili zaprepašćujući nedostatak razumevanje kod učesnika kako se pripremiti za ispitne programe i kako se obezbediti od kvara hardvera.

Računarske sveznalice

Jedna britanska studija je nastojala da ustanovi od kolike je važnosti kompjutersko predznanje kada učenik ode na univerzitet ili stupi u industriju. Istraživači su prikupili mišljenja iz obe oblasti i otkrili da akademske obrazovne instrucije više vole da imaju dake sa zdravom osnovom u matemetici i tizici nego u kompjuterskim neukama, jer su daci sa računerskom osnovom verovatno usvojili rdava navike. Industrije sa svoje strane mato značaja pridaje kvalifikaciji u kompjuterskim naukame. Računarstvo se, uz to tako brzo menje de prikupljeno znanje zastari pre nego što dak dode do diplome.

Decu opterećuju složeni problemi ako prihvetimo da je učenje programiranja progresan prislup kako prekvalifikovati naše nastavnike da daju šire, opštije obrazovanje zasnovano na računarima u kojem bi mašine bila sredstvo e ne cilj. Sveka nada de se nešto brzo može promeniti nije opravdana. Međutim, eko slagnemo remenima i odlučimo da iz loše posla izvučemo najviše što možemo prihvatanjem , kompjuterskih studija suočeni smo sa situacijom de nastavnici ne razumeju predmet dovoljno da bi ge mogli uspešno predaveti. Kako onda možemo da pomognemo deci?

Ponovno se odgovor more naći u kući. Roditelji koji su svesni opasnosti izolacije mladih mogu da urade mnogo da spreče pretvaranje svog deteta u fanatika koda. Oni mogu nastojati i na svestranijem odnosu svog deteta prema računaru. iznalazeći primene računera van školskog nastavnog programe. Pri tome se mogu suočiti sa teškim zadetkom. Programiranje je u mnogim školeme postalo snobovski predmet i dečaci koji se pripremaju za upis na univerzitet se kompjuterskim neukama visoko na svojoj listi akademskih dostignuća ubrajaju korišćenje računare (nasuprot njihovom programiranju) u glupavo računarenje. Ovaj tip omalovažavanja podmukao je i teško je izaći s njim na kraj. Ipak, dak koji je zadovoljan što koristi glupavo računarenje može biti u boljem položaju de shvati ulogu koju će veštačka inteligencije imeti kasnije u životu.

Mladi mozgovi

Pošto su istaknute opasnostr od podučavanja dece programiranju računare treba reći i to de stvaralačke mašta i sposobnost učenja eko su usmereni kako treba mogu kod tinejdžera da izbace neke odnične programere koji su u stanju da sjajno zaraduju i pre nego što završe školu. Mnoge zemlje imaju sade neku vrstu takmičenja mladih računarskih mozgove godina" kada se stvorno sposobnoj deci daje prilika da pokažu šta mogu. Pobednici ne ovim takmičenjime usmereni su se bez izuzetka na odredenu primenu računere i dobijaju nagrede jer su razvili programa koji pomažu handikepiranim osobema olakšavaju otkrivanje kriminelaca ili su od neke druge direktna koristi za društvo. Neka deca su tako vešte u pisanju programa (najčešće igara) koji privlače drugu decu, da proizvođači mikroračunare kupuju njihove programe po visokoj ceni odobrevajući im i određeni procenat od prodeje. Poznati su mnogi slučajevi 15 godišnjaka koji su zaredili mnogo novaca u svom slobodnom vremenu.

Te oboerena deca uspela su, međutim, da postenu uravnoteženi programeri sposobne de sagledaju potrebu društve za specifičnim tipom programa. Ta je sposobnost isto tako značajna, ako ne i značajnija, od čiste sposobnosti da se pišu računarski programi. Ta deca mogu biti potpuno apsorbovana u programiranju, ali je malo verovatno da bi tu aktivnost koristila kao psihološki oslonac ili kao odbrana od spoljnog sveta.

Mnogi dečaci i devojčice sa zdravim pristupom programiranju završavaju školu petkom i, namesto da se subotom angažuju u lokalnoj prodavnici poput njihovih roditelja opredeljuju se za lokalnu softver kompaniju (ili svoju spaveću sobu) da bi nastavili sa razvojem odredenog komercijalnog projekte. Treba naglasiti de te deca predstavljaju neznatnu manjinu i obično spadaju u kategoriju oboerene dece. Njihov uspeh počiva na sposobnosti de otkriju tekvu primenu računare koje će ili naći tržište ili na neki način pomoći društvu. Njihove sposobnost da stvaraju programe manje je važne i malo je verovetno da će sa napredovanjem u karijeri sami mnogo raditr na tzičkom programiranju. Verovatno je da će oni raditi u tri taze:

1 identifikecrj odredenog problema koji traži rešenje

2 analizi problema

3 skiciranju programa koji će dati rešenje.

Stvarni detaljan rad na razvoju programa biće prepušten drugim manje kreativnim programerima.

Zbog ovoga za večinu dece računarstvo treba de bude sastevni deo njihovog obrazovanja na kraju tog obrezovanje obračanje računaru za pomoć u istorijskom istraživanju ne bi predstavljalo ništa više od odlaske u biblioteku i uzimanja knjige sa polica. Deca bi bile upućana u to kako računari rade, šta mašine mogu a šte ne mogu kao i u najbolji način korišćenja raznih tipova hardvera. To bi bilo idealno stenje i do njega će se vremanom doći. Problem je jedino eko vaše dete treba da završi školu već za sledeće dva ili tri godine.

Mnoge škole koriste računere samo u nastavr matematike i nauke. Mnogo rede se oni sreću u nastavi engleskog i francuskog jezika, geografije ili istorije. de i ne govorimo o umetnosti i muzici. Postoje za to dva glavna razloge: nedostatak sredsteva i nedostatak nastavničke stručnosti. Sa pedom cene mikroračunara nedostatek stručnosti pokazaćs se kao najteži problem koji trebe savladetr.

Računari i nauka

Dobre strane korišćenje računera u nauci su očigledne. Ispitivanje životnog ciklusa stanovnika nekog jezera predmet je koji sa leko uključuje u računarski program. Program Odell Lake koji je razvile MECC (Minnesote Educational Computing Consortium) jedna od najboljih obrazovnih računarskih organizacija u svetu, omogućuje dacima da stvore model jezere i ispituju ishrambani lanac u njemu. Program je deo jednog šireg programa pod nazivom „Razumavanje lenca rshrane", sveki korisnik može de planira svoje sopstveno jezero i da uvodi razne oblike života u njega. Program trenutno razaznaje koji su oblici života inkompatibilni — na primer, pastrmke i bele riba — i o tome obaveštava učenika. U nastojanju de stvori uravnoteženo jezerce učenik će morati de prihvati neka gubitke u količini bele riba i da razmotri načine kako da reguliše broj snažnijih grabljivica.

Ova vrsta interakcionog programa stimuliše učenike da traži sve više informacija o ulozi raznih oblika života u vodenoj sredini i u zavisnosti od obimnosti programa treba de je sposoban da primi mnogo informacije direktno iz računara. Ovaj tip programa pokazao se odličnim u osvajanju i držanju pažnje učenika i praktički oslobađa učenika potrebe za nastavnikom dok stvara svoj ekvarijum. Na kraju projekta učenik stvara više ili manje uravnoteženo jezero sa svim važnim informacijama o veličini i zapremini biljnom i životinjskom svetu dobijenim u štampanom obliku.

Programi kao što je *Quake* koji takođe potiču sa MECC omogućuju učeniku da igra ulogu seizmologa i pomažu mu da odredi epicentar zemljotresa posle sekundarnog udarnog talasa. Programi se oslanjaju na raznu dokumentaciju i sastavljeni su tako da utiču i na bogaćenje đačkog rečnika i bolje razumevanje geometrije i srodnih nauka.

Nastava geografije se danas dobrim delom bavi ljudima koji žive u drugim krajevima sveta. Postoje mnogi programi koji pomažu đacima da obeležavaju u mape kretanje stanovništva i određena socijalna i ekonomska promena kao što je industrijski razvoj. Sa visokokvalitetnim grafičkim displejem deci mogu da prikažu razne delove Afrike na primer i da ih dopune senčenjem u boji o padavinama, žetvenim prinosima, gustini stanovništva itd. Pri ruci je računar koji omogućuje da se učenik u svakom trenutku prebaci sa mape na informacioni displej i tako doda do što tačnije predstave o jednoj oblasti. Tako moćni programi omogućiće da se izbrišu granice koje je obrazovanje veštački postavilo između predmeta. Dok ispituje topografiju jedne oblasti u Zimbabveu učenik će moći ako želi da doda i senčenje koje će pokazivati industrijsku proizvodnju, prosečan dohodak, zdravstvenu statistiku ili geološke informacije.

Računari i matematika

Deci su u stanju da najpre u matematici stvaraju modele problema. Matematika je možda jedna od najapstraktnijih disciplina i pruža valjke mogućnosti za dobijenje grafičkih i matematičkih modela. U svojoj aritmetičkoj ulozi računar je izuzetno koristan kad odgovara na pitanje „šta ako?" jer otkriva interakcija različitih matematičkih događaja. Koristeći popularni komercijalni program kao što je „*Visrcalc*" dak može da osnuje imaginarni biznis upotpunjen sa svim tiksnim i proizvodnim troškovima i tome dode podatke o prodaji sa odgovarajućim izdecima. Pošto je taj moćni program potpuno interaktivan, đak samo treba da menje jedan faktor — broj artikala prodatih za mesec dana, na primer — da bi saznao efekt koji bi tekva jedna promena u poslu imale na zaradu i na tinije elamente kao što je udeo tiksnih troškova koji otpada na amortizaciju vozila. Moguće je ovakva modela sačiniti i ručno ali svaki bi zahtevao mnogo časova rada. Računarski program omogućuje đaku da se beskonačno igra sa modelom svog biznisa i tako stiče temeljito razumevanje vitalnih elemenata koji zajedno proizvode dobit — ili gubitak.



U svetu mašte igra na računarima, koje su, po pravilu, često veoma agresivne, pokatkad zamenjuju potrebu mlađih za neuličkim delovanjem

Instinktivni osećaj za biznis koji se ovim dobije posle nekoliko nedelja igranja često je mnogo sigurniji od instinkta koji imaju mnogi profesionalni biznismeni koji nisu imali mogućnosti da izučavaju modele na bazi računare.

Računarske nauke kao odvojeni predmet sade su sasvim jasno definisane u većini škola i pokrivaju oblasti kao što su razumevanje elektronike programiranje i primena. Vrlo malo škole (za razliku od koledža u daljem obrezovanju) bavi se uticajem računare na društvo.

U drugim naukama — hemiji i fizici na primer — računari su dobro uvedeni i dostupni su mnogi provereni programi. Van oblasti nauke stver stoji drukčije.

Britanska organizacije MUSE (Microcomputer Users in Secondary Education — Korisnici mikroračunara u srednjem obrazovanju) trenutno ima spisak od 120 programa namenjenih školama od kojih je više od 100 orijentisano ka matematici i naukama. Postoje mnogi programi za umetničke predmete za osnovce ali mali broj za decu između 11 i 18 godina.

Muzika i umetnost predmeti su u kojima mikroračunari imaju izuzetno značajnu i posebnu ulogu. Ipak uprkos mnogim danas dostupnim pogodnostima mali broj škola koristi računare u tim oblastima.

Istraživački projekti

Sveke godina firma Commodore prikazuje kako njeni Pet mikroračunari kontrolišu jedno teško putovanje dece kroz podmukle Jorkširske močvare u severnoj Engleskoj. Uz nešto nadzora deca unose u program potrebne podatke rutu stanje okoline broj i imena učesnika itd a na komandnom mestu se zahvaljujući računaru veoma pažljivo preti tok kretanja. To je idealan tip primene koji pokazuje da uloga računara nije ograničena na jedan ugao neučne laboratorije.

Jedna škola u San Dijegu Kalifornija beleži na mikroračunaru godišnji tok migracije kitova. Iako se ovo može smatrati čisto naučnom primenom ona je obuhvatila ništa manje nego 150 đaka koji su se jednog dana posle mnogo uloženog truda ukrcali na brod da bi isplovili u Pacifik i na licu mesta videli da li njihova računarska simulacija ove najduže migracije sisara u svetu olakšava pronalaženje kitova. Šteta je da veći broj škola ne može da priušti ovaj tip stvaralaštva u rešavanju problema pomoću računara.

Ali računarstvo se brzo razvija i mada je situecija trenutno daleko od zadovoljavajuće za nekoliko godina mogla bi se ipak izmeniti tako da će deca koja uče stolarski zanat smatrati prirodnim da se obrate računaru stolarske radnje da programira ruku robota za delove koji se ponavljaju.

Šte je međutim sa decom koja završavaju školu? Neka su možda naučila nešto bajzika druga su možda predmet pratila detaljnije dok je najviše njih verovatno gurnuto u grupu koje je imale da sluša usmana objašnjenja i radi praktički nekoliko minuta u nedelji.

Oni koji završavaju školu ozbiljno se interesujući za računare mogli bi nastaviti da predmet izučavaju na kursu u nekom

koledžu ili da se direktno zaposle na računarenju. Mnogi koledži i univerziteti nude kurseve sa računarskom kvalifikacijom na kraju. Studenti koji sa uspehom završe takav kurs, verovatno će biti u stanju da u mnoštvu ponuda izaberu ono što im najviše odgovara. Bilo je poslednjih godina nekih diskusija u novinama o nezaposlenosti koja pogađa čak računarsku industriju (koja se sama često smatra prvenstvenim uzrokom nezaposlenosti). Radi se međutim o računarskom osoblju nižih kvalifikacija — kao što su programeri i operatori — dok se nastavlja (i verovatno će sa nastaviti) jaka potražnja za inženjerima sistemskim analitičarima i projektantima.

Interesantna mogućnost koja upravo postaje realna, a pruža se studentu računarstva jeste da se vrati u školu i predaje računarstvo. Do sada je ono kao specifičan predmet bilo rezervisano za univerzitete i koledže. Problem je u tome da nastavnici treba da imaju daleko širi pogled na računarstvo od ograničenog pogleda razvijenog pri radu u računarskoj industriji. Jasno je da je obrazovanje očajno oskudno u obrazovanom kompjuterskom osoblju i takva kampanja bi se nesumnjivo veoma isplatila.

Poslovi u računarstvu

U računarskoj industriji postoji nekoliko nivoa hijerarhije i mada je moguća napredovati u sistemu oni koji dolaze sa univerziteta ili sličnih institucija mogu očekivati da će napredovati brže.

Postoji šest glavnih hijerarhijskih nivoa u kompjuterskoj industriji. Najniži stepen mogao bi se nazvati „kvalifikovanim korisnikom". Ta kategorija obuhvata radnike koji su naučili da na računarima rade specijalna poslove, kao što su obrada reči ili knjigovodstvo. Često se ovaj kadar regiruje iz drugih zanimanja — na primer sekretarijata ili kancelarijske administracije — ali on uključuje i funkcije specifične za računarsku industriju kao što su operator na terminalu, operator na bušaču kartica i slično. Angažovanje na tim poslovima zahteva osnovne školske kvalifikacije i sposobnost jasnog mišljenja. Znanja kao što je rad na tastaturi normalno se uče na samom poslu.

Sledeći viši stepen jeste operator. Operatori kontrolišu rad na računaru bilo na trakama ili na magnetnim diskovima. Operatori moraju razumeti osnovu na kojoj počiva rad računara i taj tip posla često podrazumeva rad u smeni pošto mnogi veliki računski centri rade 24 časa dnevno. Poslodavci zahtevaju dobro opšte obrazovanje i izvesno poznavanja rada računara smatraju određenom prednošću. Postati operator na računaru dobra je odskočna daska za posao programera.

Programer unosi u računar uputstva na jeziku koji računar razume kao što su kobal ili fortran. Kvalifikacija potrebna da se postane programer obuhvataju jasno metodičko mišljenje i sposobnost koncentrisanja i na najsitnijim detaljima. Već programer je sasvim poseban tip osobe pa tako su za prijem potrebna odgovarajuće školska kvalifikacije prirodna sposobnost da se radi sa logikom mašine često ima veću težinu. Mnogi programeri čak ulaze u industriju bez formalnih kvalifikacija.

Programiranje je razdvojeno u dve grupe: primenjeno programiranje i sistemsko programiranje. Primenjeni programeri pišu programe da bi rešili neki specifični zadatak. Sistemski programeri pišu programe da bi držali u radu računarski sistem — takav je na primer program koji ustanovljava greške računala. Primenjeni programer je u situaciji da sreće ljude van računarske sobe — na primer stranke koji koriste računar — i da radi sa timom programera na razvijanju programa za određeni zadatak. Sistemski programeri su specijalizovaniji i skloni su ka samostalnom radu. Oni su verovatno najbliži direktnom komuniciranju sa inteligencijom mašine.

Posle programiranja industrija računala veštački povlači granicu iza koje je pristup dozvoljen samo najsajnijim programerima i najboljim diplomiranim studentima. To je carstvo sistem-analitičara i projektanata.

Sistemski analitičari

Sistem-analitičari razmatraju probleme i odlučuju kako se oni mogu rešiti pomoću računala. Neka naftna kompanija na primer otkrile je nalazište nafte ispod morskog dna. Njeni stručnjaci su izmerili veličinu nalazišta i našli da kvalitet nafte varira: neki delovi naftonosnog polja daju sirovu naftu pogodnu za proizvodnju benzina dok drugi delovi daju naftu iskoristivu jedino u proizvodnji plastičnih masa. Treba odlučiti da li vredi investirati miliona dolara u eksploataciju polja. Odluka će zavisiti od predviđanja priliva na međunarodnom tržištu nafte u toku radnog veka polja (recimo za sledećih 20 godina). Kompanija isto tako mora da odluči u kom će delu polja najpre bušiti. Pošto su investicije ogromne kompanija smatra razumnim da potraži savet od računala i pošalje mu problem da ga on analizira.

Problem razmatra analitičar. On najpre treba da prikupi informacije o ranijem stanju naftnog tržišta, zatim da se obavesti o potražnji raznih tipova sirove nafte i da konsultuje vodeće ekonomske stručnjake o njihovom sagledavanju budućnosti različitih tržišta u raznim krajevima sveta. Prikupljena su i sve dostupna merenja sa naftonosnog polja da bi se uradio računarski model koji će omogućiti simuliranje raznih rešenja za bušenje i eksploataciju polja. Analitičar sa ekonomistima i projektantima programa oblikuje potrebne računarske programe. Konačni rezultat će se verovatno sastojati od velikog broja programa (koji možda uključuju grafičku predstavu strukture polja) koji pažljivim oblikovanjem i dedukcijom mogu pružiti logične zaključke o mogućnostima eksploatacije polja. Izvršni organi naftne kompanije biće u stanju da se na ovom modelu igraju pitanja „šta ako?" otkrivajući kako bi razne odluke o cenama, tehnikama rafinisanja i tržišnim pristupima uticale na čitavu sliku. Dolazeći lako do svih informacija koje su im potrebne oni će moći da donesu konačnu odluku o najboljem načinu eksploatisanja svog naftonosnog polja.

Ovo je uprošćen primer. Stvarni projekt može zahtevati nekoliko godina rada angažovanje velikog računarskog tima i zasnovanost na detaljnim računarskim predviđanjima. Produkcija polja i perspektiva svetske privrede iz primera sa ipak može videti da je uloga analitičara i projektanta programa vitalna jer računari mogu da rade samo u okviru unapred propisanog metoda i na osnovu primljenih informacija. Analitičar je odlučujući faktor u dobrom korišćenju računara, kada računar stvarno postaje dopuna i proširenje čovekovog mozga.

Ima i nekih drugih važnih poslova u industriji računara, mada ih je malo koji se tako visoko cene kao posao analitičara. Izuzetak je, možda, projektovanje hardvera. Postoje dobri izgledi za elektronske inženjere svih nivoa od centara za popravku autoputeva do istraživačkih odeljenja ali oblasti razvoja proizvoda i čistog istraživanja otvorena su samo onima sa najvišim kvalifikacijama u elektronskom inženjerstvu.

Mnogi analitičari i projektanti mašina i softvera napreduju do rukovodećih i konsultantskih položaja, ali te titule često znače samo to da pojedinac ima više vlasti vilo često radeći u sopstvenoj režiji. Radni sadržaj posla često nije promenjen.

Ako industrija računala cveta mnoge druga industrije podižu recesiju, jedan OECD izveštaj objavljen krajem 1982 ukazuje na to da će i nekoliko sledećih godina ta navolja pritiskati većinu zapadnih zemalja — sa značajnim izuzetkom SAD i Japana.

Mikroračunarski centri

Neke vlade posebno britanska, smatraju da mikroelektronika može pomoći da se reše neki kratkoročni problemi nezaposlenosti. Po jednoj shemi koju je ustanovila britanska vlada da bi omogućile sticanje izvesnog radnog iskustva velikom broju nezaposlenih mladih (pre nego postanu nezaposlivi), nudi se mesto u 40 mikroračunarskih centara u Britaniji. U tim centrima mladi koji su završili školu mogu pohađati kurseve o raznim aspektima mikroračunarenja primajući uz to malu novčanu naknadu. Ta shema pruža dvostruku korist onima koji su prošli kroz školu bez dodira sa računarom (jer su napustili školu pre nego što je računar stigao ili nisu bili izabrani da ga koriste) pruža mogućnost da se s njim upoznaju a poboljšava i njihove izglede da nađu neki posao.

Može se postaviti pitanja kako osposobljavanje jednog završenog đaka da dobija zaposlenje koje počiva na računaru (čije je uloga verovatno da zameni nekoliko manuelnih radnika) pomaže problemu nezaposlenosti. Dvosmislenost o ulozi računara u zapošljavanju podsaknula je izveštajima koji predviđaju masovnu nezaposlenost kao direktnu posledicu računarske automatizacije. Jedan nedavni izveštaj kazuje da će nezaposlenost izazvana automatizacijom dostići 16% u Zapadnoj Evropi. Za one koji zavišavaju školu bez ikakve računarske pismenosti izgledi da se zaposle zaista su ništavni. Mnogi stručnjaci smatraju da mikrotehnologija pre povećava jaz između onih koji imaju i onih koji nemaju nego što dovodi do veća jednakosti, kao što neki futurolozi predviđaju.

Bez obzira na svoje sposobnosti i interesa, deca koja se približavaju dobu kada završavaju školu zaslužuju naše simpatije. U obrazovanju u sustini jednom informaciono-prelaznom procesu, ona su se našla zahvaćena informacionom revolucijom na koju se mnogo lakše adaptiraju od svojih učitelja. Po završetku obrazovanja oni se suočavaju sa jednim svetom koji se i sam u prelaznom periodu između pred- i posle-računarskog doba našao u meteżu. Ona moraju biti spremna da pređu u tom svetu e jedna od bitnih kvalifikacija te spremnosti jeste sposobnost da vladaju kompjuterskom inteligencijom i da je koriste.

nemojte plakati za eplom!

Najzamršenija i najuzbudljivija kompjuterska igra koja se tranutno igra zove se kupovina računara. Ona je obavijana misterijom, opasnostima, intrigama i prijatnim pozivima. Kao i u svakoj pustolovnoj igri, glavni igrači — bez obzira da li traže izgubljeno Blago, Malteškog Sokola ili Pravi Kompjuter — pitaju se: Kome mogu da verujem? To je pitanja o kome zaista vredi razmisliti!

Kupiti ili umreti

Pre svega, na verujte nikome ko tvrdi da vam je računar potreban *odmah*. Fanatici kompjutera su uavojili nešto žara — i neke od izraka — religioznih zanesenjaka „Računajtal Kraj je blizu!"

U stvari, nema potrebe za žurbom. Živeli ste dugo bez računara i moći ćete sasvim tino još malo. Jedno od prvih pravila u kupovini kućnog računara jeste da se na žuri. Nema potrebe da kupite računar još ove nedelje. Kupovina kućnog računara zahteva dosta strpljenja, izdržljivosti — i vremena.

Ne verujte oglasima za računara. Oglašavanje se pretvorilo od nenamerna samoparodije iz sedamdesetih godina u nadrealizam u osamdesetim. Jedan tekući oglas koristi slogan „Daleko od očiju — daleko od pameti"

Ne verujte takode, ni časopisima. Neki specijalizovani časopisi za računara svakim danom postaju sve sličniji haj-faj magazinima. Jesta li ikad pročitali loš izveštaj o velikom proizvodaču (čitaj — oglašivaču) stereofonskih uredaje u nekom od haj-taj časopisa? Prikazi su toliko laskavi i toliko puni obzira da au potpuno beskorisni.

Kao i haj-faj časopisi, i magazini za računara zavise od prihoda od oglasa za ista proizvode o kojima objavijuju prikaza. Ako su suviše često iskreni — dovidenja oglašivači I, na kraju, zbogom časopise! „Popular Computing" ja odbio da objavi jedan moj prilog o tome zašto je Appla ila slaba vrednost. Razlog? Prema izjavi urednika „Desilo se da je voiim Appie Compjuter korporaciju" Mislim da je dozvoljeno zamisliti da on, takode, voli i mnogo atrana Apple-ovih oglasa u boji svakog meseca.



Karikatura Milos Artović

Vapaj u divljini

Dakle, koma verovati? Prijatajlime koji imaju računara? Ne obaveznol Ljubitalji računara su obično skloni da se podaju onoj vrsti kompjutara koji poseduju bez obzira da li stvarno vredi ili na. Sem toga, nakon što potroše na baš zanemarijivu sumu novca da bi kupili računar, malo ljudi će reći, „Pogrešio sam. Trebelo je da uzmem nešto drugo". Više ljudi se kuna u svoj „ševroiet" nego što ga prokinje.

I dok smo kod teme kolsi prodevcime računara možeta verovati otpriiiika isto toliko koiiko i prodavcima polovnih automobila. Ponekad razmišljam da su se svi nezaposieni prodavci automobila prijaviii i odmah dobiii zaposlenja u prodavnicama računara od jedna do druge obala (misli se na Atlantski i Tihi okean).

Ima, doduše, i nekoiiko dobrih prodavaca računara. Oni znaju isto tako dobro kao i ja da većina daja lošu reputaciju manjini. Budimo far — isuviše stvari se dešave previše brzo da bi se blio ko održao, a da se na zapusti. Ja se, možda, na mogu zapustiti, ali je na moram da čekam na muštarija osam časova na dan.

Zato, za sve na svetu, na uaimajte moju reč ni za šta. Ja sam samo glas koji vepi u divljini (Los Angeiss). Misli iznesene u ovom čisnku nisu ništa drugo nego moja subjektivna, pristrasna, izuzetno lična mišljenja.

Koma dskia, verovati? Ako želite da imeta pravi računar za pravu cijenu —

Jugoslovenski ljubitelji računara okrenuti su, uglavnom, engleskom i nemačkom tržištu. U domovini kućnih računara, Americi, na čitav fenomen i, rekli bismo biznis, gleda se sa znatno manje nervoze. Koje je njihovo viđenje kućnih računara i pojedinih zvezda na računarskom nebu? Članak Pitera Viljamsa (Peter Williams) o najpopularnijim modelima kućnih računara „sa druge strane velike bare" radikalan je čak i za američke prilike, ali može da pomogne i našim ljubiteljima računara da raščiste sa nekim predrasudama. U svom članku Viljams propušta kroz sito i rešato četrdesetak modela računara — mi smo izbor suzili samo na ona koji su poznati i u našoj zemlji.

Imajte poverenja u sebe! Uronite, učite koliko možete razgovarajte se što je moguće više sveta i na kraju ćete razaznati ono što će vam biti korisno

TIMEX SINCLAIR 1000

Timex Sinclair je poznat kao prvi svetski računar za jednokratnu upotrebu. Kupite jedan za oko $ 49-95, odnesete ga kući, koristite nekoliko dana i odlučite: (A) računari vam se dopadaju i želite da dobijete neki bolji ili (B) računari vam se na dopadaju i ne želite da imate više ništa se njima. U oba slučaja oslobađate se Timex Sinclaira se otprilike isto toliko ceremonije kao kada praznite mišolovku.

COMMODORE

VIC 20 i Commodora 64 predstavljaju izuzetnu vrednost među jeftinim kućnim računarima. Za manja od 100 dolara VIC 20 je mnogo bolji računar za jednokratnu upotrebu nego Timex Sinclair. Za oko 200 dolara Commodore 64 ima skoro ave hardvarske uređaje kao i Apple IIe, koji košta 1395 dolara. U grupi kućnih računara se cenom ispod 500 dolara Commodora 64 je jasni pobednik.

TEXAS INSTRUMENTS 99/4A

Gde li su samo našli ime kao što je 99/4A? Izgleda da je čak i Bil Kozbi (Billi Cosby) bio nesposoban da spase ovu mašinu. Texas Instruments ju je početkom ove godine (1983) prodavao uz gubitak, nadajući se da će razliku nadoknaditi prodajom programa. Plan nije proradio. Iskazan je veliki gubitak. Akcija su pala. Ova situacija nije mnogo iznenadila one koji prate računara Texas Instruments je obeshrabrio sve druge da rade softver za 99/4A. To je jednako pametno kao da obeshrabrite Standard Oil da proizvodi benzin za vaš automobil. Ako T I ne promeni tu politiku, i to brzo, neće još dugo biti na tržištu kućnih računara.

ATARI

Uzduž i popreko ATARI zemlja, ljudi hodaju okolo sa staklastim pogledom, mrmljajući „Šta se to dogodilo? Za trenutak je izgledalo da je Atari zagospodario tržištem kućnih računara. Koštao je manje od Apple-a. Imao je „pakmena" imao je primamljiv alogan („Jeste li igrali Atari danas?") Šta se desilo? Desila se konkurencija. Coleco je ponudio boilja igra — Commodore je ponudio jeftiniji računar. Ako Atari na izvuče nešto magično iz svog šešira, i to uskoro, može završiti na sve većoj listi kalifornijskih ugroženih vrsta.

APPLE IIe

Apple računar ne predstavlja dobru vrednost već čitave dve godine. I Apple II i Apple iii su prekupi, u poređenju se silično opremljenim računarima, još od 1980. Apple IIe (Apole kaže da „e" znači poboljšani-anhanced, ja kažem da to znači skup — expansive) dobio je još nekoliko znakova, mala slova, malo više memorija i priključak za štampač, smanjan mu je broj čipove (smanjenje na predstavlja vrednost za kupca, ali čini računar jeftinijim za proizvođnju) i povećana cena. Čak i ja, koji sam imao malo poverenja u Apple, mislio sam da će ponuditi više a naplaćivati manje.

Glavni razlog zašto ljudi kupuju Apple IIe ja softver i paritarijska oprema koju proizvoda drugi proizvođači. Većina toga potiče iz kasnih sedamdesetih godina kada je Apple bio jedini računar u boji na tržištu. All Apple gubi te prednosti. Postoji više lepih računara koji rade se softverom Apple-a.

U stvari ako pogledate na spisak od 16 000 programa koji stoja na raspolaganju za Apple I koji je formata telefonskog imenika, videćete da je njihove velika većine bezvredna izgledaju kao prilozi na konkursu za upla u višu školu („Imeta tri seta na raspolaganju da napišeta program za računar. Počnite ')

RADIO SHACK

Radio Shack je bila jedna od prvih kompanija (druge su Commodore i Appie) koja je ponudila lična kućna računara. Nakon nekoliko dosadnih godina u kojima je izgledalo da se Radio Shack, kao i Apple, odmara na lovorikama (Appls se više na odmara — on se ucrvljao) izišao je sa nekoliko računara koji su dobri i jednim koji je vrlo dobar.

Dobra vest kod Radio Shacka je model 100. Možda na bista želali da vodite poslovanja svoja kompanija na njemu, ali bista mogli požalati da ga stavita u avoju ručnu torbu. Trsetetura je puna veličina i ima velilku pregiednost. Ekran je tečni kristel, kao kod džepnih računara i lapisuje, osam redova od po 40 karaktera. Nlja ono što bi nazvali mašina za obredu teksta — više je mašina za zapis teksta. Sve što pohranita u model 100 moža se pranesti na drugi računar za naknadno izdavanja iii reviziju. Podaci se smeštaju u vratu RAM koja nikad na zaboravlja. Nakon što se podaci pranesu iii lakorista, mamorija se može obrisati es naredna zadatke.

IBM PERSONAL COMPUTER

IBM je segradio izvanredan kućni računar. Umesto da „sklapa" mali računar koriščenjem malo od ove IBM mašine a malo od one, IBM je sastavio grupu inženjera, dizajnera, programara i, da nekoliko zaljubljenika u kućne računara, i izdvojio ih se zadatkom da projektuju i izrede IMB PC.

Među proizvođačima malih računara postojao je strah da će IBM progutati tržište kućnih računara. Umesto toga, izgleda da je IBM stvorio čitavu novu industriju. Svako ja dobio dso učešća: proizvođači periferijske opreme, prodavci programa, firma za prodaju putem pošte i preprodavci. IMB lapoljava neobičan stepen susretljivosti prama ljudima koji izraduju stvari koje se ugrađuju u, na ili priključuju za IBM PC. Pra ovoga, politika IBM-a ja bila: ako to nije napravio IBM, ne treba ni da se koristi za IBM. Sada je IBM čak objavio shamu PC-a i nudi svu moguću pomoć.

Čak, kao čudo neđ čudima, IBM prodaja proizvod druge kompanija pod imenom IBM. To je jedan Epson, proizveden u Japanu, na koji se dodaja pločica IBM-a. To je toliko radikalno kao da Steinway stavi avoja ima na klavire proizvedene u Japanu. S druge strana, IBM/Epson izgleda malo star uz nove Epson štampaće.

Tastatura je jedinstvena. Naki ljudi ja vola a neki mrza IBM tastatura ja dizajnirana se onim što ja poznato pod nazivom „audio taktilni tidbek". Kada pritisnete tastar, vi čujeta i osetite klik.

Drugi čudan alament tastatura ja ameštaj SHIFT tipke. Godinama, proizvođači malih kompjutera, želeći da s ponosom istaknu kvalitet avojih tastatura, navodili su da su selectric-tipa". Tipka su bila na istim pozicijama kao kod IBM Selectric-a. IBM iz razloga poznatih jedino atvarasou ili atvarsocima, nija aledio selectric tormat. All, Keytronica ja već predstavio tastaturu koja sa može priključiti na IBM I ispraviti ave ta uh, ekscentričnosti. To ja lepa stvar se IBM-om: on će najverovatnija postati lični računar sa najviše podrške u istoriji. U stvari, on to već sada jesta.

Kao i kod Apple IIe, za sve što želita da radita se sopstvenim računarom, bez aumnja ćeta naći parifarijaki uređaj iii softver koji će vam to omogućiti. Apple IIe ipak, to redi na nivou više škola, a IBM za — odrasla.

Da li bista varovali, ćak i pra godinu dana, da će IBM ikada prodaveti proizvod koji se zove XT? Zvuči kao ime sportskog automobila iii poboljšanog uređaja za pranja usta. Kako brzo veliki momci dozvoljavaju da im izmakne ugled kada treba „napraviti" novac. I tako to ide. Dobra vesti i loše vesti o avelu kućnih računara su da se on brzo menja. Niko ne može biti stalno u toku. Sad uzmite malo slobodnog vremena da izadeta napolja i dodirnete robu.

Namojte plakati za eplom!

čekajući QL-a



Mašina na koju se more računati QL

Nikada varovatno, nećemo saznati šta je Klajva Sinklera navelo da istrči pred rudu sa svojim novim računarom — preterano poverenja u svoje razvojne inženjere pritisak javnosti koja je žitavih četrnaest meseci sa nestrpljenjem iščekivala njegov novi potez žalja da na samom početku godine preduhitri konkurenciju koja je počela da najavljuje nova mašina ili sa već pojavila sa njima ili zapravo poslovna nesmotrenost — nitl je to za čitavu priču previše bitno Tak pred nama se iako je od njegove januarske promocija prošlo više od pet meseci još uvek nalazi samo desetak istih fotografija i more nagadanja iz engisske štampe o računaru čiji razvoj, po svemu sudeći neće biti okončan još nekoliko meseci. Zato se ovaj tekst neće previše baviti detaljima računara QL — može sa dogoditi da se oni promene pre nego što se na našem izdanju osuši štamparska boja

Više „hop" nego skok

Pošto je na januarskoj promociji zašao među publiku i napunio svoj šešir čekovima na po 400 funti Klajv Siklar ja prestao da se pojavljuje u javnosti Za to ja imao dobrih razloga — isporuka prvih mašina sa toliko otegla da su čak i Englezi koji su spremni da se zakunu na ZX znak počeli da sumnjaju da će se ovaj računar ikada pojaviti na tržištu Početak isporuka sa zakašnjenjem od gotovo četiri meseca malo ja umirio računarske duhove ali nija ni iz daleka otklonio ave dileme Računar se isporučuje sa jednom ružnom izrastinom na zadnjoj atrani kutija koja treba da otkloni neka konstruktorska zabluda iz prvih dana i nedovršenim operativnim sistemom Šta uostalom misliti o računaru koji nakon nekoliko meseci upotreba treba vratiti tabrici na doradu?

Za nagli skok na polju kućnih računara koji po mišljenju njegovih tvoraca, donosi QL mogao ja da buda primeran samo neki egzotičan termin poput „quantum lsep" iz kvantne fizika (kvantni ill nagli skok — otprilika kao svetionik na sred beskrajnog pustog okeana) Za ovaj skok trebalo bi najviše da buda zaslužan mikroprocesor 68008 firme Motorola, koji Sinklarovi stručnjaci za propagandu daklarišu kao tridesetdvobitnu mašinu To ja praktično, bila prva stvar na kojoj je Sinklar bio uhvaćen i koja ja nagovestila potonja komplikacija sa QL-om lako inženjeri ne uspevaju da se slože šta je to mikroprocesor od 32 bita sasvim ja sigumo da 68008 to nija — njegova unutrašnja arhitaktura ja latina u formatu od 32 bita ali magistrala za podatke ima samo osam linija To u suštini znači da on operiše sa podacima kao i bilo koji osmobitni procesor Na račun Sinklarove pratancioznosti Englezi zbijaju šalu kako QL zapravo, ima samo dva bita — jedan spreda i jedan pozadi! lako se radi o veome brzom i veoma moćnom procesoru od naglog akoka je izgleda ostalo samo jedno „hop"

Ko mnogo ima . . .

Oko drugog procesora iz QL ima znatno manje dilema Radi se o mikrokompjutaru na jednom čipu sa oznakom Intel 8049 u kojoj ja porsed procesora ugradena i sistemska i radna mamorija! Njemu su poverani trivijalni ali česti zadaci — kontrola tastature ganarisanje zvuka i prijam sanjskog signala — čima ja glavni procesor rastaređen jedna valika i dosadna briga U svome radu glavni procesor se dosta oslanja i na dva specijalno projektovana čipa — jedan kontroliše akran i mamoriju a drugi mikrodrajvova časovnik lokalnu mrežu i amitovanja serijskih signala — koji preuzimaju dosta velika teret štedeći snagu glavnog gazda za udarna zadatka I to ja kad ja reč o operativnoj i kontrolnoj logici, ava!

Za sistemsku memoriju u prvobitnoj verzlji računara bilo je odvojano 32 K ali sa pokazalo da ja operativnom sistamu na tom prostoru previše tesno Prvi primarci isporučeni su sa dodatkom od još 16 K ROM-a U konačnoj varzlji, ovaj dodatak će biti ugraden u računar čime će se QL po sistamskim bajtovima približiti američkim mašinama čiji su konstruktori u ovom detalju oduvek bili široka ruka Pošto će uz čip više koji nija ukaikulisan u cenu računara dobiti i neke neplanirana pogodnosit ovaj previd vlasnicima QL na bi trebalo previše da kvari raspoloženje

Prilikom dimenzionisanja radna memorije konstruktori QL au rakio bi sa bili znatno manje škrti — 128 kilobajta u koja odjednom staja devedesetak atranica teksta, mogu sa visanicima računara „galaksija" učiniti „dovoljnim za čitav Beograd" Pokazuja se međutim da ko mnogo ima mnogo i troši da bi razvio svoju punu snagu operativnom sistemu QL-a nija dovoljan postojeći RAM i moraće po svemu sudeći da sačeka na mamorijako proširenja od 500 kilobajtai lako sa konstruktorima QL može uputiti zamerka za neuskladenost mašinskog i programskog dela računara potraba za obilnom mamorjom pra predstavlja vrlinu nego nedostatak To je u suštini jedini način da se ostvari novi kvalitet

Ipak nešto drugo

Q tastaturi računara QL nija se praktično znalo ništa ava dok prve mašina nisu isporučana kupcima Stručnjaci za propegandu au nedorečenim podacima vešto podgravali utisak da sa Sinklar ovoga puta opredelio za mehaničke tastera i profesionalna performansa. Pokazalo sa, međutim, da ja i ova tastatura zasnovana na principu membrana (i) ali i da ja to uradeno dovoljno solidno za profesionalnu primanu računara — brz i tačan unos podataka u mašinu Prilikom nedavna posata Galaksiji' Anđelko Zagoreiec se žalio da ja tastatura previše meka, osetljiva i neposlušna".

Januarska prašina nad računarom QL je već odavno pala, ali se još uvek ne vidi ništa. Mikroprocesor sa tridesetdvobitnom arhitekturom, 128 kilobajta radne memorije, novi bejzik koji sa arhaičnim mikrosoftovskim varijantama ima zajedničko samo ime, novi operativni sistem koji pretenduje da postane standard za tridesetdvobitne procesore, dva ugrađena mikrodrajva, četiri besplatna programa i ostale pogodnosti za svega 400 funti dovele su u iskušenje čak i najbolje poznavaoce računara, koji počeli ja na da se odriču svojih skupocenih računarskih sistema. Jutro je, međutim, pametnije od večeri — vreme je počelo da pokazuje da je u QL ugrađeno više lepih zelja njegovih tvoraca nego čipova i sistemskih bajtove.



Tačke na i ' ili kamen oko vrata. Ugrađivanje mikrodrajvova u sam računar predstavlje hrabar i rizičan potez koji se još uvek nije srećno završio

dok su kritičari časopisa 'Your Computer' došli do zaključka da se na njoj radi sa lakoćom. U svakom slučaju, na njoj se uz ave standardna znake i široku razmaknicu nalazi pet tastera opšte namena kojima se mogu dodeliti proizvoljna funkcije. Naredbe se, kao i na svim ozbiljnim računarima, na unose prako tastera sa dodeljenim funkcijama (tzv. keyword) nego se kucaju slovo po slovo, kao i bilo koja druga reč.

Fina grafika . . .

Najočigledniji iznenađenja međutim vezana su za ekran QL-a. Zamišljen veoma ambiciozno: on je organizovan tako da vlasnicima nove Sinklerove mašina obezbedi pogodnosti znatno skupljih sistema. Pored fine grafike 256×256 tačaka se osam boja QL može da radi sa ultrafinom grafikom 512×256 tačaka u četiri boje. Grafički modaliteti se postavljaju automatski — u zavisnosti od toga da li se nakon uključivanja računara odbera rad sa televizorom ili rad sa monitorom. Korišćenjem pogodnosti za crtanje lečkama — vertikalnih i horizontalnih pruga i velikih i malih tačaka — na ekranu se može proizvesti 255 različitih boja! Boje i oblici slova su kažu veoma stabilni u obe grafička modaliteta — ni naitik lelujanju kod računara Spectrum.

Ekran je — prvo u nizu iznenađenja — podeljan na tri dela. Dno ekrana — zelena slova na crnoj pozadini — rezervisano je za četiri programska linije. Leva polovina ekrana — bele pozadina sa crvenim slovima — predvidena je za listing programe a desna — crvena pozadina sa belim slovima — za njegovo izvršavanje. Unošenje programa i naknadnih izmena je olakšano ekranskim editorom e tastiranje dovedeno do savršenstva — dok sa prako leve strane ekrana kreće listing na desnoj se može pratiti njegovo dejstvo. Na žalost ove pogodnosti nisu izbrušena do kraja. Kada korisnik pokuša da dveput izlista naki program — e jedan program sa u toku razvoja iista bezbroj puta — u belom prozoru doiazi do 'zagušenja'. Osim toga podele ekrana na tri dela predstavlje uobičajeni način rada QL-a u monitorskom modalitetu i ne može se nt na koji način preskočiti osim direktnim komandama. Da bi ekran dobio uobičajeni izgled, traba otkucati najmanje dve naredba što očigledno, nije previše elegantno.

. . . i softverske smicalice

Mogućnost da se na ekran postevi više prozora, pri čemu se u svakom mogu dešavati različite stvari predstavlja jednu od najudernijih pogodnosti novog Sinklerovog operativnog sistema. Časopis 'Your Spectrum' je međutim nazive običnom šalom i softverskom — smicalicom. Umesto čvratog sistema kakav na primer koristi Apple, Digital Research ili Microsoft tvorci QL-a su, naprosto po ekranu iscrtali okvira. Za svaki okvir dodule obezbedena je programska kontrola sa evim parametrima kao i za čitav ekran. All ona ne dejstvuje pouzdano kada na može de stane u jedan tekst „ledno" prelezi i u drugi prozor — brisanjem jednog prozora isto sa može obrisati i onaj do njega. Slični efekti zaključuje 'Your Spectrum' – mogu se postići i na bilo kom drugom računaru uključujući i četiri puta jeftiniji Spectrum.

Procesor
Motorola 68008 na 7.5 MHz (unutrašnja arhitektura 32 bita, magistrala za podatke 8 bita, direktno adresira 1 megabajt radne memorije)
Sistemska memorija
32+16 K se operativnim sistemom QDOS i superbejzikom — može se proširiti do 64 K
Radna memorija
128 K od čega je 32 K zauzeto za ekran — može se proširiti do 640 K
Operativni sistem
QDOS se sledećim prednostima: multiprogramski rad, sistematizovanje zadataka po prioritetu, prozori na ekranu, rad sa periferijskom opremom nezavistan od vrste uređaja
Jezik
Superbejzik sa sledećim prednostima: strukturisano programiranje, jednostavno uvođenje novih naredbi (uključujući i sintaksu), brzina izvršenja nezavisna od dužine programa, jednostavno pozivanje mašinskih programa, jednake pogodnosti za reči i nizove, ekranski editor
Ekran
Ultrafina grafika na crno-belom ili monitoru u boji u dva modaliteta — 512×256 (četiri boje) i 256×256 (osam boja) — i 25 redova teksta sa od 85 (monitor) odnosno 40 do 60 znakova u svakom redu (TV)
Tastatura
QWERTY punog formata sa 65 tastera, širokom razmaknicom, levim i desnim šift tasterom, četiri tastera za kontrolu kurzora i pet tastera opšte namene
Mikrodrajv
Dva ugrađena mikrodrajva minimalnog kapaciteta 2×100 K, prosečnim vremenom pristupa podatku od 3.5 sekundi i brzinom učitavanja od 15 K u sekundi
Proširenja
Pored priključaka za napajanje, TV i monitor, obezbedeno je i devet portova za periferijske uređaje: radna memorija, mikrodrajv, sistemska memorija, serijski interfejs (2), lokalna mreža (2) i palice za igru (2)
Serijski interfejs
Dva standardna RS232C interfejsa za printer, modem itd. sa podesivom brzinom prenosa između 75 i 19200 bita u sekundi, za dvostruki dvosmerni prenos
Dimenzije
136×472 mm
Masa
1388 G
Cena
399 funti

„Dva časa nasamo"

Ovim sa ekološki izmedu QL i Spectruma ni iz daleka na iscrpljuju — njihovi operativni sistemi i njihovi bejzici imaju, očigledno, zajedničke temelja, koji bi upornog računarskog 'arheologa' verovatno odveli sve do računara ZX 80. Nije razume se nimalo jednostavno auditi o programskom jeziku jednog računara na osnovu spiska naredbi i funkcije i šturih tudih utisaka tipa „dve časa nasamo sa QL-om". Programski jezik QL-a, nazvan superbejzik, pisan je očigledno se velikim ambicijama i snažnom željom da nadmaši eve svoje prethodnike. Mada pomalo peti od kembridžskog eksdemizma — dugačke nezgrapne naredbe i šturа jezgrovita poruka — superbejzik izgleda ima eve odlika zrelog programskog jezika: višelinijske funkcije, procedure sa lokalnim promenljivim, promenljive sa dugim imenima, formatirano listanje i ekranski editor. Arhaične naredbe tipa GOTO i GOSUB uvrštene su izgleda samo iz nostalgije a FOR-NEXT petlje liče na kiasiku samo po imenu.

Da bi se otkrile dobre i slabe strane superbejzika potrebno je barem nekoliko meseci rada se QL-om, ali je već sada jasno da mu nedostaje malo više logike i finoće u detaljima. Uočena su takođe i neka veoma uspele i moćne rešenja. Procedura se mogu izvršavati i potpuno nezavisno — izvan svog matičnog programa. Pošto svaka procedura može imati i svoja parametre uvođenje novih naredbi u superbejzik predstavlja pravu dečiju pesmicu. Fiskalibilnost očigledno čini jednu od jačih strana ovog jezika. Na cenovu hardverskih specifikacija mogao bi se pomisliti da se programi izvršavaju „kosmičkom brzinom". Prvi utisci, međutim, govore da je QL samo dvaput brži od Spectruma, a u FOR-NEXT petljama ni toliko.

Biser u školjci . . .

Konstruktorima QL-a po svemu sudeći, najviše problema zadaja operativni sistem nazvan QDOS. Iako je prošlo već više od pet meseci od najave računara on ni do danas nije završen, niti ima izgleda da se to dogodi u nekoliko narednih meseci. Zamišljen možda previše pretenciozno — da postane standard za Motorolinе tridesetdvobitna procesora — QDOS trenutno najviše hramlja upravo u onim osobinama koja treba da predstavljaju njegovu najjaču udarnu snagu: promena parametra u toku izvršavanja programa na primer vrednosti neka sistemske promenljive paralelan rad više programa u memoriji i na ekranu i udoban i brz rad sa mikrodrajvovima.

Nedorečenost, verovatno predstavlja osnovni razlog koji je naveo tvorce QL-a da operativni sistem sačvore za korisnika. QDOS se na žalost na može koristiti direktno u sopstvenim programima — za sada sa iz superbejzika na može prići nijednoj od njegovih udarnih pogodnosti. Deo za rad sa diskovima ima dovoljno snage sa dva mikrodrajva sa po desetak fajlove na svakom, ali se može pokazati potpuno nemoćnim kada se jednoga dana na QL priključi tvrdi disk.

Rad se tastaturom, ekranom i periferijskim uređajima zasnovan je na sistemu atruja i „kanala" koji je razrađen na računaru ZX Spectrum (vidi str. 68—71). Ovaj sistem omogućuje veoma jednostavnu i elegantnu softversku vezu sa periferijskim jedinicama — takvu da na priključeni uređaj QL gleda kao na organski deo svoje sopstvene mašine. Povezivanje određene atruje u mašini sa određenim kanalom (periferijski uređaj) može se izvršiti sa čitavim spiskom parametara koji se, na primer odnose na vrstu štampača i modelitet rada. Otvaranjem atruje u memoriji se automatskt rezerviše prostor za 16 K alfanemskog programa za pogon priključenog uređaja. Sistem je čak u stanju da prepozna da li je periferijski uređaj sa aukcesivnim (štampači, tastatura, mreža) ili slučajnim pristupom (mikrodrajv, tvrdi disk) podacima. Vreme će, možda, pokazati da su atruja i kanali ubojitija QL-ovo orude od multiprogramskog reda.

. . . a školjka na dnu mora

Mogućnost paralelnog rada više programa u memoriji — dok čekate da na printeru

Rečnik QL-a

Procedure		MRUN	PAPER	INT	IF THEN ELSE
		NEW	RECOL	LN	DEFine
ARC	OPEN	RETRY	SCROLL	LOG 10	PROCedure
ELLIPSE	OPEN–IN	RUN	STRIP	SIN	RETURN
LINE	OPEN–NEW	SAVE	UNDER	SQRT	DATA
POINT	POKE	$BYTES	WINDOW	TAN	END FOR
SCALE	POKE–W	SEXEC	Funkcije	PI	END IF
BEEP	POKE–L	STOP	BEEPING	CHR$	END REPeat
INPUT	RANDOMISE	AT	INKEY$	CODE	END DEFine
PAUSE	CLEAR	BLOCK	KEYROW	PEEK	END SELect
PRINT	CONTINUE	BORDER	ABS	PEEK–W	EXIT
BAUD	EXEC	CLS	ACOS	PEEK–L	NEXT
CLOSE	EXEC–W	CSIZE	ACOT	RND	
COPY	LBYTES	CURSOR	ASIN		
COPY–N	LIST	FLASH	ATAN	Naredbe	
DELETE	LOAD	INK	COS	FOR	
DIR	LRUN	OVER	COT	REPeat	
FORMAT	MERGE	PAN	EXP	SELect ON	

izlista vaš najnoviji program, računar vas može zabavljati lepom partijom šeha — sa svim tokovima na ekranu trebalo bi da predstavlja bisar QDOS operativnog sistema. Nevolja je, jedino, u tome što je ta školjka još uvek duboko na dnu mora. Paralelni rad uostalom nije neobičan čak ni sa najjaftiniji računar. Svaki operativni sistem radi paralelno na različitim zadacima — očitavanja tastature, prikazivanje znaka na ekranu, generisanja zvuka — ali je obično toliko spor da izgleda kao da se ovi poslovi odvijaju jedan za drugim. Oni se, da stvar bude lapša, tako i odvijaju. U čemu je onda štos kod QL-a? On ima dve procesora. Prelazak sa jednog programa na drugi je, osim toga, dovoljno brz da se to ne primeti. Multiprogramski rad značajno podiže kvalitet obrada podataka. Voleli bismo, zato da tvorci QL-a istraju u svojim namerama. Time bi se u kućna računare presilila još jedna pogodnost velikih računarskih sistema.

Kamen oko vrata

Nijedan hardverski detalj na QL-u nije izazvao toliko protivurečnih reakcija kao mikrodrajvovi. Dok jedni njihovu ugradnju u sam računar smatraju mudrim, ključnim potezom, drugi su ubeđeni da oni predstavljaju QL-u kamen oko vrata i da će Sinklar kad-tad morati da potraži drugo rešenje. Mnogi su još i sada ubeđeni da će u konačnu verziju QL-a biti ugrađeno nešto sasvim drugo — minidiskovi firme Hitachi. Zagovornici mikrodrajvove amatraju ovaj Sinklarov izum dosta elegantnim aredatvom za trajno akladištenje podataka, a protivnici neozbiljnom igračkom koja ni po čemu nija primerena mogućnostima QL-a.

Varoveli u mikrodrajv ili ne, pokazuja se da on zasad predstavlja najveću kočnicu razvoju novog Sinklarovog računara. Da bi sa na traci pronašao određeni podatak, računar radi u prazno 3,5 sekunde, a učitavanja jednog od besplatnih programa traja sedamdesetak sekundi — trenutak u poređenju sa kasetom, večnost u poređenju sa diak jedinicom! Pretarana sporosti su, u zadnji čas, postali i avesni i tvorci QL-a, koji grozničavo tragaju za efikasnijim operativnim sistemom. Tragajući za podatkom na traci operativni sistem ida od jednog do orugog bloka — ako ono što se traži nija u prvom bloku, prelazi se na drugi, treći i teko ave do kraja. Direktan pristup blokovima će varovatno ubrzati učitavanja podataka, ali ni to po svemu sudeći, neće biti dovoljno da se spase postojeća koncepcija.

Neko to voli drugčije

Na zadnjoj strani QL-a nalazi se nekoliko priključaka — računska mreža, monitor, TV, serijski interfaja, palice za igru, apotjanji ROM i apoljašnji RAM. Ma kako to zvučalo neverovatno, u mašinu koja je između ostalog namenjena i za obradu teksta nija ugrađan paralelni interfaja za štampač niti se našlo mesta za port opšta namene! Sinklar, koji se očigledno, upinje iz petinih žile da nametna avoja atandarde računarskom svetu pokazuja, izgleda veliku odbojnost kada se radi o atandardima koja au izmislili drugi.

Uz računar QL isporučuja se i četiri ozbiljna „protiv" programa koja dolaze iz radionica firme Psion — program za obradu taksta, za obradu podataka, knjigovodstvo i poslovnu gratiku. U sapadnom svetu gde ja odnos prama programima znatno drugačiji nego kod nas, jedan ovakav sottverski paka! predstavlja sam za aaba dovoljan razlog da se nako odluči za kupovinu QL-a. Mada imaju veome efektan izgled ekrana, oni au pisani u tranutku kada ja operativni sistam QL-a bio u ambrionalnom atadijumu i sasvim sigurno ne mogu da damonstriraju najvažnija mogućnosti računara. Na treba međutim aumnjati da će vrlo brzo na tržišta biti izbačene unapređena varzija.

Performanse računara QL objavljana au šest meseci ranija nego što je trabalo i konstruktori sada imaju velikih problama da održa datu reč. U tome možda leži i najveći nedoatatak QL-a — od njega se očekuja znatno više nego što može da pruži. A brzoplietost, kao i svuda, ima avoju cenu — jedan inača i savremeno i zrsio koncipiran računar izlazi pred avoja kupos sa nalzbrušenim bajzikom, nedovršenim operativnim aistemom i nakotiko otvorenih pitanja oko konačna verzija mašine. Isporuka računara ja, izgleda, bita ista toliko brzoplata kao i njegova promocija — pored avih nevolja Sinklar sada ima za petame i nazadovoljna visanika čija mašine ne rade kako treba.

„Ko igra na oštrici tehnološkog noža, može i da sa poseče" — mudruju Englazi. A Klajvu Sinkleru preostaja de varuja u tzraku de sa ipak najalade ameja onaj koji to čini poslednji. Sudeći prama prvim ocenama da „QL sasvim zavreduja avoju cenu" i „da će izvršiti anažan uticaj na razvoj kućnih računara, sličan onome koji au ostvarili Appla, IBM PC i Spectrum", možda će ipak on biti taj.

Jova Regasek

Elektron

Ejkornu nije bilo nimalo lako da koncipira kompjuter koji bi sa uklopio u liniju koju su činili Atom, BBC A i BBC B: novi računar ja morao da bude toliko jeftin da ugrozi modele iz ekonomska klasa, toliko moćan da konkuriše Commodoru 64 i drugim računarima iz sradnja klasa, a ipak na toliko moćan da uzdrma pozicija BBC modela A i B. Svi ovi zahtavi, jasno, nisu mogli da budu ispunjani, pa ja Ejkorn odlučio da žrtvuje BBC A i izbaci na tržišta računar koji će u mnogoma kopirati BBC, ali koji će biti lišan nakih skupih delova i tako biti na raspolaganju onima koji za kompjutar nisu spremni da izdvoje viša od 200 funti. Electron je ispunio očakivanja iz našag prvog specijalnog izdanja: došao je na sam vrh tabela najtraženijih modela na avropskom tržištu!

Dlaka u jajetu

Electron ima 32 Kb ROM-a i isto toliko RAM-a ROM je kao i kod BBC B podeljen na operativni sistem (ekvivalentan sa verzijom 1.20) i bajzik (BASIC II naravno) Umesto bajzika jasno može da se priključi EPROM sa nekim drugim programskim jezikom (za sada je na raspolaganju fort) procesor taksta iti nešio treće I pored ovih očigledniih sličnosti hardver Elacliona je sasvim originalan I nama većih dodirnih tačaka sa BBC-jem Upotrabljeno je manje komponenti i ostavljeno manja praznih podnožja za proširanja, računar ja tako postao manji jeftiniji i na žalost sporiji

Glavna zamerka koju korisnici BBC-ja obično upućuju Electronu je nedostatak telatekst kompatibitnosti Nija nikakva tragedija što računar nija kompatibilan sa teletekst mrežom već što mu nedostaje MODE 7 koji većina korisnika BBC B (računajući i autora ovoga taksta) upotrabljava otprilike 90% vremena rada Zašto je mod 7 važan? Jednostavno zato što računar dok radimo u njamu rezerviše samo 1 Kb RAM a za video-memoriju pa je 31 Kb slobodan za naš program i podatke Korisnici Elektrona obično rade u modu 6 koji za ekran rezerviše čitavih 8 Kb

Electron ima generator tonove koji u svemu imitira BBC-jev osim u jednoj sitnici u jednom momentu možemo da generišemo samo jedan ton što na prvi pogled značajno smanjuje njegovu vrednost Čini nam se da zaključke na traba donositi preuranjano treba biti pravi majstor da bi se sinhronizovao rad BBC-jevih kanala i dobija orkestarska muzika U najvećem broju programa (čak i izvanrednih akcionih igara poput Snappara) i BBC B konstiti samo jedan kanal za sintezu tona!

Razlike bez značaja

Ostale razlike izmedu BBC-ja i Elactiona su primatna praktično na pivi pogled tastatura je da počnamo od nja manja i sastoji se od znatno manje tastera (55 prema 74) Korisnici Electrona au na prvi pogled ostali bez tastera kojima mogu da dodeljuju proizvoljna iiterala Kažemo na prvi pogled, jer sa istovremenim pritiskom na CAPS LOCK i naki od brojeva dobija funkcija akvivalantna pritisku na naki od BBC-jevih narandžastih tastera Kao posebna pogodnost, taster BREAK takode može da doblja dodatne funkcija

Tastar BREAK je jedna od velikih BBC-jevih i Elaclionovih pogodnosti Ma šta računar radio u nekom trenutku (izvršavanje bajzik ili mašinskog programa potpuna blokada usled nake greške traženja programa sa kasete ili diska ili nešto četvrto) pritisak na BREAK ga vraća u normalno stanja i reinicijalizuje bajzik Program koji se nalazio u računaru biva privremeno izgubljen i viaća se tek kada odtkucamo OLD Zašto bismo kucali OLD kada god prilisnamo BREAK? Lakše je pia početka rada otkucati KEY 10 OLD CTRL M i tastar BREAK će pored uobičajena funkcije izvršaveti i OLD po rainicijalizaciji računara

Loša strana Electrona je što praktično nema priključenih interfejsa Na njegovoj zadnjoj strani su samo priključci za kasetofon televizor i kolor odnosno crno-beli monitor dok je na donjoj strani samo **običan port za ekspenziju**. Svi oni koji žele da kupuju periferijska uradaja moraju dakle da nabavljaju razna dodatka kojima ćemo posvatiti sledeće redove

U prodaji je takozvani teletekst adapter koji Electronu dodaja mod 7 sa svim opcijama — izgleda da je tetatekst mod podržan programom u ROM-u i da je potraban samo hadver kojim bi se realizovao Programi za podršku štampača su takoda u ROM-u, dokaz je što au naredba VDU1 — VDU 3, kod BBC-ja namenjene kontroli štampača označene kao Rasarvad u uputstvu za upotrabu Da bismo dakle povezali Electron sa štampačem traba da kupimo Centronika Interfejs kojt će nas koštati tridesetak funti Povezivanje sa diskovima je takoda moguće ali je potrebno kupiti disk Intertejs kojr se prema naproverenim informacijama smešta u posabnu kutiju i povezuje sa portom za ekspanziju Čak je i za dodavanje palica za igru potrebno dokupiti A D konvertor Jasno je da svi ovi dodaci značajno povećavaju cenu Electrona koja postaje sve približenija BBC-javoj

Razmislite još jednom

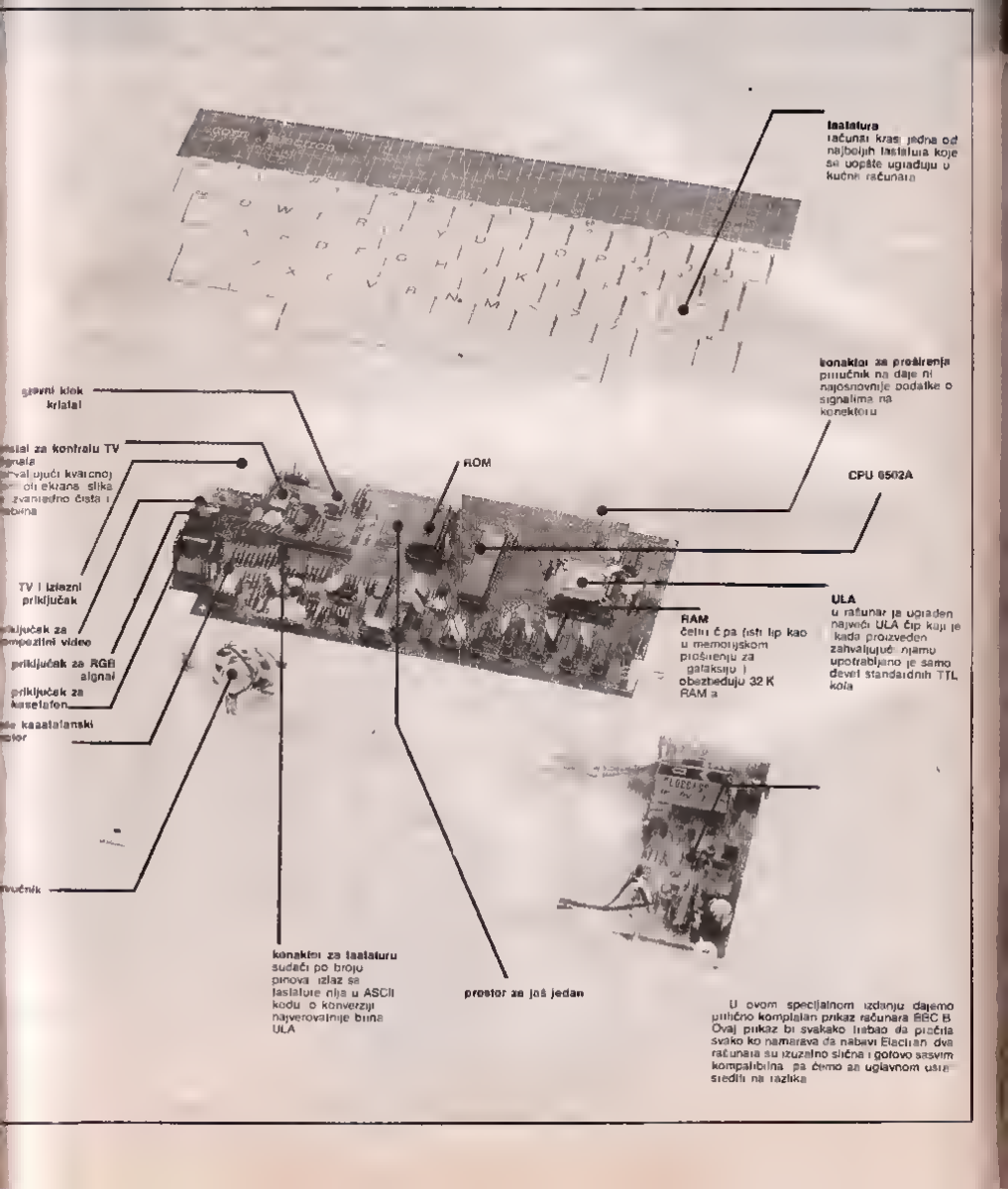
Programska podrška je jedna od jačih strana Ejkornovog novog kompjutera Praktično svi programi pisani za BBC B (a takvih je kao što smo rek dosta na tržištu) mogu da se p Electronu i obratno Jedini problem je mod 7 koji je zbog mogućnosti pisanja velikih slove korišćen za naslovna strana mnogih BBC-jevih igara Izvršavanja ovakvih programa na Electronu je naravno moguće ako se iišimo prizora koji obično traje dok sa program učitava sa traka I kod nas su u modi zajednički aranžmani kupovina stranih programa koje ravnopravno i u međusobnoj saradnji nabavljaju vlasnici BBC-ja i Electrona Vlasnicima Electrona je osim Acorn Usara i A&B Computing na raspolaganju i izvanredni časopis Elbug koji izdaja klub Beebug.

Sve u svemu Ejkorn Electron predstavlja izvanredan izbor Nije lako naći računar koji košta 200 funti sa moćnim i brzim strukturiranim bejzikom grafikom 640×256 mogućnošću sintaza tonova i (eventualno) glasa i visoko profesionalnom tastaturom Preporučili bismo ipak onima koji razmišljaju o kupovini Electrona da realno procene svoja potrabe za ekspanzijom sistema Ukoliko shvatite ava ograničenja računara i zaključite da vam ona na amataju kupita Electron što pre Ako sa druge strane želite proširenja biča svaka ko bolje da razmotrite mogućnosti nabavke duplo skupljeg BBC B — ne samo što ćete proći jeftinije nego ćeta imati manjih problema sa uvozom Filozofija kupovina osnovne varzije računara i postapenih proširenja prama potrebama ja za strance preporučljiva ali kod nas vladaju drugi zakoni

Dejan Ristanović

Acorn Electron

CPU 6502
Klok: 1·79 MHz
Memorija 32 K ROM-a i 32 RAM-a
Ekran: do 32 reda sa po 80 znakova 8 boja 127 definisanih i 255 karaktera koje može definisati korisnik
Interfejsi TV kompozitni video RGB video kasetofon sistematski bas
Jezici bejzik sa ugrađenim asemblerom
Tastatura: 56 mehaničkih tastera opcioni unos naredbi jednim tasterom 10 tastera za dodeljivanje funkcije
Dokumentacija izvanredna sa dobrim opisom operativnog sistema i obiljem detalja za programiranje na bejziku i mašinskom jeziku
Dimenzije 340×160×65 mm
Cena 199 funti



U ovom specijalnom izdanju dajemo prilično kompletan prikaz računara BBC B. Ovaj prikaz bi svakako trebao da pročita svako ko namerava da nabavi Electron dva računala su izuzetno slična i gotovo sasvim kompatibilna pa ćemo se uglavnom usredsrediti na razlike

BBC

U specijalnom izdanju „Računari u vasoj kući" BBC mikrokompjuter je ocenjen kao najkompletniji stoni računar koji se može naći na tržištu. Ta konstatacija je izrečena posle opsežna konsultacija stranih časopisa, ali sa autor ovoga teksta u nju do kraja uverio tek kada je dobio prilika da intenzivno radi sa BBC-jem i upozna njegova tajne. Šta je, dakle, to što BBC B (model A sa praktično viša i na proizvodi) čini najboljim?



Računar bez premca BBC B

Prvi utisak o BBC-ju more da buda povoljan na (za jedan stoni kompjuter) velikoj ali veoma lepo oblikovanoj kutiji dominira visoko profesionalna tastatura na kojoj su osim standardnih alfanumerika zastupljeni mnogi specijalni znaci (računajući ćak i velike i srednje zagrade ali začudo ne i znak copyright koji ostaje privilegija vlasnika Spectruma) Utisak o kompletnostr testature dopunjava desal narandžastih tastara kojima možemo da dodeljujamo proizvoljna literala Imamo utisak da jedino tastature Hewlatt-Packardovih kompjutara mogu da se mara I (verovalno) nadmaše BBC-jevu

Ekranski editor

Po uključenju računara nalazrmo se u takozvanom modu 7 Još smo u Računarima 1 namenrekti da je kod BBC kompjutara napravljen kompromis izmedu potrabe za visokom rezolucnom r velikim slobodnim RAM-om tako što su uvedeni raznr gratički modovi — ukoliko želrmo visoku rezoluciju moramo da se pomirimo se time da ćemo imatr malo RAM-a I obratno Mod 7 na ekran troši samo t krlobajt pa je verovatno zato i izabran za početak rada To je dobro ali se pojavljuje jedan drugi po našoj ocenr za početnike vrlo ozbiljen problem mod 7 je kompatibilan sa takozvanim teletekstom dok ostali sadrže karaklere oblikovane po ASCII standardime To značr da na primer taster na kome je ispisana srednja zagrada u modu 7 daje levu strelicu šlo u prvi mah može ulekako da zbunr Zato bismo početnrku svakako savetovali da pra nego što započne razgovor sa računarom izabere MODE 6

Čim počnu da se pišu prvr bejzik programl ukazuje se potreba za upoznavanjem editora koji omogućava ispravljanje grešaka u programu BBC može da se podići ireobičnim ali izuzelno moćnim i fleksibilnim ekranskim editorom Pre nego što ga opišemo naše smeteli da pomenemo razlike izmedu ekranskih i linijskih (linijski editori se koriste kod velikog broja računara iz ekonomske i srednje klase) editora da bismo ispravili neku naredbu Ireba de olkucamo EDIT (rli neku odgovarajuću naredbu) i broj trnije Zatim pomoću teve i desna strelice pomeramo kurzor t, odgovarajućim komandama vršimo ispravke posla kojih pritiskamo ENTER Kod ekranskih edrtora su u upotrabi I druga dve strelice pomoću njih pozicioniramo kurzor na naku poziciju akrana i vršimo ispravke „na licu mesta Samo se po sebi razume da su ekranski editori daleko tlaksrbljnlji t moćniji, ali oni lmaju veltku manu ako smo na prtmer kucali naku dužu komandu (naredbu bez linijskog broja) i prt tom pogrešlr ostajemo jedino da je ponovo kucamo osim ako želimo da se pomrimo sa naredom i konfuzijom na akranu

BBC-jev editor je u osnovr ekranski ale olklanja ovaj nedostatak čim pritisnemo neki od četlri tastara na kojima se nalaze strelice standardnr kurzor se pratvara u beli kvadrat r pojavljuja se drugi kurzor koji sa daljim prttscrma na stralice pozicionira na proizvoljnu poziciju akrana Pritiskom na taster COPY sa ta pozicija kopiramo tekst ne poziciju kurzora-kvadrata (taster COPY kao r svi drugi tasteri lma funkciju autorepeat ako ga duže prilskamo) Na taj načm se može kopirati proizvoljna grupa slova — nljedan tekst se akrana nije mrtav ako je kompjuter na primer, negde ispisao Mistake možamo slobodno da tu reč uklopimo u naku našu naredbu ili

komentar. Samo se po sebi razume da je moguće kopirati i proizvoljne komende što posteje posebno korisno kada poželimo da snimimo isti program nekoliko puta

Strukturirano programiranje

Po upoznavanju editora možemo be početnu sa pisanjem prvih bejzik programa i upoznavanjem BBC-jevog programskog jezika. Ubrzo uvidamo koliko je on kompleksniji i moćniji od ostalih verzija bajzika BBC bejzik je naime sakupio u sebi mnoga dobre svojstva drugih programskih jezika, ponajviše paskala. Tu pre svega mislimo na velike potencijele za pisanje strukturirenih programa. Strukturirano programiranje omogućava na samo jednostavnije pisanje programa sa manje grešaka nago i jednostavne docnije modifikacije programa i njegovo prilagodavanje drugim kompjuterima

Strukturirani program se piše po tekozvanim stepenima složenosti: početni problem se deli ne nekoliko etapa, e zatim se svaka te etapa posebno programira deijim raščlenjivanjem na jednostavnije probleme. Postupak se nastavlje sve dok se na dođe do krajnje jednostavnih problema koji se rešavaju u nekotiko programskih linija. Svaka etapa rešavanje problema se izdvaje u poseban potprogram koji je u principu nezavisan od ostatih i ima svoje ulazne i izlazne veličina. Pošto je taj program (nazivamo ga i „modul") razvijen i potpuno testiran, snimamo ga na traku i pretazimo na sledeći. Na kraju posle jednostavno učitavamo module i povezujemo ih u celinu. Kada je na primer docnije potrabno da prilegodimo već postojeći program drugom računaru, biće dovoljno da baz ikakvih intervencija u glavnom programu. prereidimo module koji operišu sa tastaturom i ekranom

Da bi se na racionalan način pisati strukturirani programi, treba imati mogućnost pozivanja potprograma se prenošenjem argumenate i definisanjem lokalnih promantjivih. Ovekva mogućnost je BBC-ju dete preko naredbi PROC FN i LOCAL U glavnom programu se na primer. može naći naredba PROCsuma (A B C). Kada računar naide na nju potražiće naredbu DEFPROCsuma (X Y Z) (umesto X, Y Z mogu, jasno da se koriste i bilo koje druga slova ili reči) i promenljivoj X dodetiti vrednost promentjive A e promenljivima Y i Z, respektivno vrednosti promentjivih B i C Zatim se izvršavaju naredba iza DEFPROCsume. Ukotiko je prva od njih naredba LOCAL A u potprogramu će biti nezavisna od promenljive A u glevnom programu, što znači de će potprogram biti sasvim autonoman Umesto procedure može da se koristi i funkcijski potprogram koji počinje sa DEFFNsuma (A B) e poziva se na primer sa X=2+FNsuma (total, sabirak)+32

Ništa nije savršeno

Iako su konstruktori BBC-jevog bejzika očigtedno uložili veliki napor be omoguće pisanje sevremenih atrukturiranih programa očigledno su propustili de poseo dovedu do kreje Pre svega prenošenje matrica i nizove keo argumenate nije jednostevno i što je još gore nije dokumentovano Tek se uvidom u stručne časopise ulazi u ovu tajnu i tako pisanje potprograma za rad sa matricama i sistemima jednačine (škoiski primar korisnosti strukturiranog progremiranje) posteje moguće No rzgleda de nikakvim trikovima nije moguće sintetiseti tokatnu ON ERROR naredbu neredbu pomoću koje biamo bili u stenju de kontrolišemo eventualne greška u proceduri a de ne rametimo tretirenje grešaka u drugim procedureme i glevnom programu

Nedosteje oaim toga tokalne RESTORE naredba kao i mogućnost de se jednostavno sačuve stetus računere (sadržeji svih sistemskih promenljivih koje su ineče, prilično razasute po memoriji) koji bi se povretio po nailesku na standardnu naredbu ENDPROC Ove nedostatka 90% korisnika neće ni zapaziti ali nikome, aigurno neće biti pravo što u obilju naredbi MERGE nije nešto mesto Ova naredba se naravno može jednostevno sintetiseti eli pogotovu za početnika to nije isto kao de je ona u ROM-u Pri povezivenju modula neobično je korisno što računar ima vrlo tteksibilnu i izuzetno brzu naredbu RENUMBER

Ultravisoka rezolucija . . .

Rezotucija 640×256 sasvim jasno pokazuje da su Ejkornovi inženjeri želeti da BBC ima moćnu grefiku pa su se potrudili da obezbede kompleten set komandi za kontrolu displeja Lepa je pogodnost što je mapa ekrana iata u svim modovima jasno je na primer da u modu 1 ima dupto manje tačaka koje mogu de se kontrotišu nego u modu 0 ati je tačka sa koordinatama 1279 1024 uvek u levom gornjem ugtu pe će program planiren za mod 0 rediti i u modu 1 iako će jasno slike biti grubije

Za crtanje se koriste naredbe MOVE i DRAW čije funkcije postaje jasna kada pročitamo njihove nazive No, mnogo je korisnije naredba PLOT koja ima tri argumente prve dve govora o tunkciji koju treba izvršiti (PLOT 4 je, na primer isto što i MOVE). e preostaie dva deju koordinate tačke. Naredba PLOT obavlje nake poslove koji bi ineče mučiti programera crtanje isprekidena linije bojenje kvadrata trouglova i mnogouglova i slično Autoru ovoga tekste se posebno dopale paralelna mogućnost epsolutnog i reletivnog (preme adresi zadnje crtane tačke) adresirenja gratičkih jedinica svaki od ovih metode ima svoje prednosti pa je lepo ako su oba prisutne

U prospektime se obično navodi da BBC poznaje 16 boja, što je samo detimično

INSTRUKCIJE RAČUNARA BBC I ELECTRON

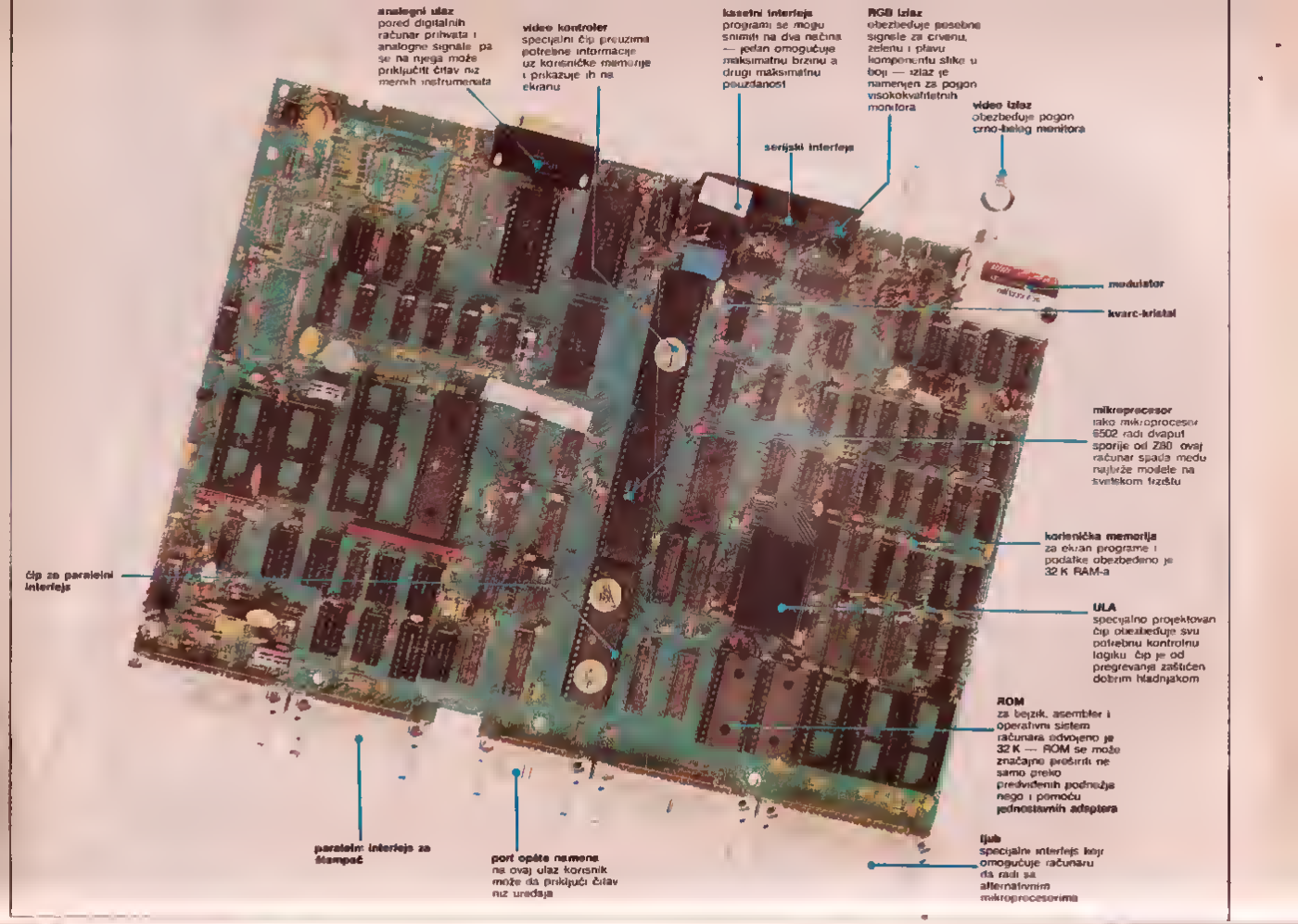
ABS	ACS	ADVAL	AND	ASC	ASN	ATN	AUTO
BGET#	BPUT#	CALL	CHAIN	CHR#	CLEAR	CLOSE#	CLG
CLS	COLOUR	COS	COUNT	DATA	DEF	DEG	DELETE
DIM	DIV	DRAW	ELSE	END	ENDPROC	ENVELOPE	EOF#
EOR	ERL	ERR	EVAL	EXP	EXT#	FALSE	FN
FOR	GCOL	GET	GET#	GOSUB	GOTO	HIMEM	IF
INKEY	INKEY#	INPUT	INPUT#	INSTR	INT	LEFT#	LEN
LET	LIST	LISTO	LN	LOAD	LOCAL	LOG	LOMEM
MID#	MOD	MODE	MOVE	NEW	NEXT	NOT	OLD
ON	OPENIN	OPENOUT	OPT	OR	PAGE	PI	PLOT
POINT	POS	PRINT	PRINT#	PROC	PTR#	RAD	READ
REM	RENUMBER	REPEAT	REPORT	RESTORE	RETURN	RIGHT#	RND
RUN	SAVE	SGN	SIN	SOUND	SPC	SQR	STEP
STOP	STR#	STRING#	TAB	TAN	THEN	TIME	TO
TOP	TRACE	TRUE	UNTIL	USR	VAL	VDU	VPOS
WIDTH							

tačno: različitih boje ima u stvari 8 dok se preostalih osam dobije kombinecijom osnovnih Osim toga nema smisle rediti u modu 2 (prevsiko zauzeće RAM-a) — botje se opredeiiti za mod 4 iii mod 5 i imeti ne raspolaganju 4 boje tepu rezoluciju i više stobodnog memorijskog prostora Ni u ovim modovima medutim nismo ograničeni na boje koje je Ejkorn izabreo, što je jedna od najlepših finesa na BBC kompjuteru primenom naredbe VDU može se bito koje od četiri stanberdne boje predefinisati u neku drugu. U modu 0 na primer raspolažemo samo crnom i belom bojom ali posle VDU 19 1 4 0 bele boje postaje zetene!

. . . i kontrola ekrana

Naredba VDU je neobično moćne pa se oni koji sazneju da je VDU X isto što i PRINT CHR$(X), obično nemaio iznenade Zaista VDU je uvedeno jedino zbog jednostevnijeg i brzeg pisanje i uvek može da se zameni sa PRINT CHR$. Kodovi 0—31 su rezarvisani za specijelna namena od kojih je jedna (promena početnih boja) već pomenuta Pomoću VDU ae de pomenemo samo neke primene jednostavno definišu „prozori" za tekst i gratiku (na taj način korisnik može da radi ne jednom delu ekrana i time na kvari izgied ostatka na kome je, recimo neka slika), menja grafički koordinatni početak uključuje i isključuje kurzor menja njegov oblik aktivira i isključuje štempač VDU 25, na primer, potpuno imitire već dovoljno moćnu naredbu PLOT C X Y

Mikroprocesor: 6502A
Klok: 2 MHz
Memorija: 32K ROM-a sa bejzikom i operativnim sistemom i 32 K RAM-a
Ekran: osam različitih grafičkih modova sa maksimalnom rezolucijom 640×256 tačaka, do 32 reda sa po osamdeset znakova i do 16 boja
Interfejsi: televizor, crno-beli i monitor u boji, disk-jedinica, printer, analogni ulaz, port opšte namene, port za vezu sa drugim mikroprocesorima
Jezici: bejzik i asembler
Jezici na raspolaganju: lisp, fort, logo
Tastatura: 74 mehanička tastera (64 standardne namene i 10 za dodeljivanje funkcija)
Dokumentacija: odlična, sa detaljnim prikazom rada i programiranja mikroprocesora 6502, ali previše napredna za početnike


analogni ulaz
pored digitalnih računar prihvata i analogne signale pa se na njega može priključiti čitav niz mernih instrumenata
video kontroler
specijalni čip preuzima potrebne informacije uz korisničke memorije i prikazuje ih na ekranu
kasetni interfejs
programi se mogu snimiti na dva načina — jedan omogućuje maksimalnu brzinu a drugi maksimalnu pouzdanost
RGB izlaz
obezbeđuje posebne signale za crvenu, zelenu i plavu komponentu slike u boji — izlaz je namenjen za pogon visokokvalitetnih monitora
video izlaz
obezbeđuje pogon crno-belog monitora
serijski interfejs
modulator
kvarc-kristal
mikroprocesor
iako mikroprocesor 6502 radi dvaput sporije od Z80 ovaj računar spada među najbrže modele na svetskom tržištu
korisnička memorija
za ekran programe i podatke obezbeđeno je 32 K RAM-a
čip za paralelni interfejs
ULA
specijalno projektovan čip obezbeđuje svu potrebnu kontrolnu logiku čip je od pregrevanja zaštićen dobrim hladnjakom
ROM
za bejzik, asembler i operativni sistem računara odvojeno je 32 K — ROM se može značajno proširiti ne samo preko predviđenih podnožja nego i pomoću **jednostavnih adaptera**
paralelni interfejs za štampač
port opšte namene
na ovaj ulaz korisnik može da priključi čitav niz uređaja
ljub
specijalni interfejs koji omogućuje računaru da radi sa alternativnim mikroprocesorima

Set naredbi za crtanje dopunjavaju GCOL i COLOUR za biranje boja. Naredba GCOL, osim standardne funkcije omogućava da se crtež logički sabira (funkcije AND OR i XOR) sa onima što već postoji na ekranu i tako formiraju vrlo složene slike. Za igre i obradu teksta je korisno i definisanje karaktera koja se jednostavno postiže jednom od opcija naredbe VDU. Što je najlepše predefinisani karakteri mogu da budu i na tastaturi pa ee na primer srednja zagrada može zameniti našim čevom č.

Sintesajzer bez tastature

Generalisanje tonove je jedna od najjačih karakteristika BBC-ja. Četiri kanala (jedan samo za generisanje buke) omogućavaju kako sintezu komplikovane muzike teko i proizvođenje glasa (meda sa u BBC ugrađuje posebna opcija u vidu sintetizatore glasa koje ugiavnom dodaje potrebne softver). Jednostavna nota se dobijaju naredbom SOUND koja ima četiri parametre: broj kanala, visinu note trejanje i jačinu zvuka. SOUNO 1—10 101, 10 deja, na primer, četvrtinu note C u „Moderato" ritmu. Pravi efekti sa madutim dobijaju primenom naredbe ENVELOPE pomoću koje definišemo odvojnicu tona koji najpre može de sa pojačava (modulacija može da bude i po trankvenciji e ne po emplitudi) smanjuje opat pojačeva i teko ponavlja dok sa ne sinhronizuje sa tonom na nekom drugom kanalu ili, jednostavno dok korisnik na pritisne taster Escapa.

Dok se naredbama za generisanje elike zaista na mogu uputiti nikakve zamerke naredbe za generisanja tona su izuzetno moćne ali je autor ovoga teksta stekao utisak da su neverovatno složene i da najveći deo korisnika (računajući i njega) nikada neće uspeti da iskoristi sve njihove potencijale. Trabalo je da budu formirane naredba koja će biti prilagođena potrebama korisnika, a ne čipu 76489.

Operativni sistem

Konstruktori računare BBC su težili i uspeli da on bude neobično brz i efikasan. BBC B premda ima star osmobitni procesor 6502, radi dalako brže čak i od većine šesnaestobitnih kompjutera što samo potvrđuje tezu o revnoprevnosti hardvera i softvera pri konstrukciji računara. Jasno je da su morali de budu učinjeni neki kompromisi (ne mnogo njihi) da bi sa dobilo u efikasnosti. Jeden od tih kompromisa je red sa stringovima ukoliko na početku reda programa definišemo maksalmalnu dužinu sitenumerika, utrošak memorije će biti minimalni. Računar neće imati ništa protiv produžavanje tih sitenumerike u toku rade. eli će tada utrošak memorije biti skoro katastrotelen.

Iskusniji korisnici će, svekako poželeti de rada sa mašinskim jezikom i de iskoriste ugrađeni asembler. Firme se potrudile de taj posao što više olakša dobrom dokumentacijom. Ugrađeni asembler je podržan bejzikovim editorom, što znači de se esemblerske rutina pišu poput bejzik programa čim računar nelda na otvoreno srednju zagredu počinja sa esembiirenjem i smešta dobijeni sadržaj u memoriju počevši od lokacije na koju ukazuju promenljiva P%. U sora programu se jasno koriste labele komentari i slične esemblerske naredbe.

je rad sasvim komforan. Pozivanje mašinskog programa se postiže naredbama USR i CALL koje imaju mnogobrojne opcije za prenošenje parametare u potprogram.

U okviru uputstva za upotrebu date su početne adrese svih važnijih rutina operativnog sistema. Tako je onaj ko piše mašinska programe pošteden brige o crtanju i pisanju po ekranu, radu sa kasetofonom, generisanju tonove. Namerno smo rekli „adresa rutina operativnog sistema a ne „adresa rutina bejzik interpretatora" — ova dva programa su naime sasvim odvojena.

Operativni elatem zauzima 16 Kb i obuhvata sve rutine neophodne za kontrolu računera. Bilo je mnogo verzija operativnog elatema što stvare nebrojena problema sa kompatibilnošću. Ako kupujete BBC B obavezno proverite da li je u njega ugrađena verzije operativnog sistema 1.20i. Ukoliko računar, kade otkucate *HELP, odgovori nekim drugim brojem slobodno potražite drugog prodavca!

BBC-jev bejzik takođe zauzima 16 Kb i emaštan je u jedan EPROM 27128 (sada se u računara etevljeju i ROM-ovi što varovatno znači de je razvoj operativnog sistema okončan). Naposredno pored ovoga EPROM-e sa nalaze još tri podnožja — u njih korisnik može de emsati druge EPROM ili ROM čipove. Programe u njime jasno možemo i sami da razvijamo ali će sa većina korisnika opredeliti za gotova. Ukoiiko na primer kupimo ROM sa paskal kompajlerom otkucaćemo *PASCAL, po uključivanju računara i bejzik će biti isključen ne zauzimajući nikakav adrsani prostor. U podnožja na morau da se stavljaju samo interpretatori novih programskih jezika — mnogi koriste prosesor taksta ili neki drugi uslužni program kao alternativu bejziku. Ovi programi pogotovu ako imate BASIC II (proverite to tako što ćete pritisnuti BREAK i otkucati REPORT ako računar ispiše (C) Acorn 1983. imate drugu verziju bejzika i vaš računar poznaje mnoge naredbe koje nisu spomenute ček ni u vašem uputstvu za upotrebu!) mogu da pozivaju programe iz bejzik interpretatora premda proizvođač nije ponudio potrebnu dokumentaciju.

Drugi procesor

BBC B ima ogromne mogućnosti ekspanzije. Na njegovoj zednjoj strani su pored standardnih priključaka za televizore RGB i crno-beli monitor i kasetofon ugrađeni dvanasatobitni AD konvertor RS423 interfejs (verzije čuvenog RS232 serijskog interfejsa) i priključak opcionog Econet interlejsa (dokupljuje se kao posebni dodatek i služi za povezivanja više BBC računare u mrežu). Sa donje strane kutije su portovi za printer diek-jedinica (diak interfejs se posebno dokupljuje ali nije mnogo skup) tube interfejs (za izuzetno brzu komunikaciju 6502 sa drugim procesorom) — 1 MHz bus za povezivenje sa leboratorijskom i drugom specijalnom opremom i stendardni „korisnički port" univerzalne namena.

Posaboo je interesantno dodati drugi procesor obzirom da se tako rešave najveći problem BBC-je — nedostatak RAM-e. Za sada može da se kupi još jedan 6502 sa 64 KB RAM-a (njegovim dodevenjem sa gotovo neverovatno ubrzeva izvršavanja bejzik i mešinskih programa jer soltver podržave multiprocesoraki rad) i Z80 procesor sa istim RAM-om koji daje CP/M kompetibilnost. Čini nam se, ipak de u ovom

momentu na treba trošiti nekih 200 funti za kupovinu drugog procesore. Bolje ja sačekati do leta u nadi da će se prama obećanjima pojaviti nova opcija National semiconductors 16032 mikroprocesor sa šeneasstobitnim basovima i tridssetdvobitnom arhitekturom koji će moći de opetužuje 16 megabajta RAM-a. Opremljen njime BBC B će postati toliko moćen kompjuter da će teško biti pravazidan bilo čima osim canom u sledećih nekoliko godina i svakako dovoljan za sve programe koje požalite!

Mogućnost dodavanja drugog procesora ima na žalost i negativnih posledica. Da bi programi bili kompatibilni sa proširenim kompjuterima ne traba nikako poukovati ' direktno u video mamoriju već se traba opredeliti za stalno pozivanje potprograma iz ROM-a. To na bi bilo ništa strašno da firma Ejkorn nije zaključila de će najbolje spročiti korisnike de to rade ako uskrati potrebne informacija. Tako u uputstvu za upotrebu nalazimo adrsae sistemskih potprograma ali ne i objašnjenja „pakovenje" video memorije, tačnu mapu RAM-a i sistemskih promenljivih, podršku direktnog pristupa portovima. Korisnicima ostaje jedino da sami pribave potrebne informacija što se nerevno obavlje preko klubova programare.

Klubovi programera

Pomiljnjem klubova programera dolazimo do poslednjeg (nikako na najmanje važnog) poglavlja ova priče o BBC B — programske podrške. Za BBC je napisano zaista mnogo izvanrednih uslužnih sistemskih i, naravno, zabavnih programa. Igra su neobično efektne i teško da se izgled ekrana igera za druge računare može meriti sa BBC-jevim. Za ovaj računar se najviše pišu akcione igre (brz je i ima odličnu grafiku i ton) dok je aventura znatno manja najviše zbog nedostataka RAM-a. Logičkih igara takođe ima i to izvanrednih — šahovski program White Knisht (Beli kon) je jedan od najboljih sličnih programa za stone računare uopšte! Od akcionih igara posebno vredi nabaviti Sneppera (virtouzno urađena verzija pakmena) a od aventura — Snowball i, naravno, Hobbite.

Možemo da vam preporučimo dva komercijalne časopisa A&B Computing (izlazi šest puta godišnje) i Acorn User naročito ovaj drugi (pak prave stvar je tek učlanjenje u kompjuterski klub Beebus. Ovaj klub može da sa podiči sa preko 30000 članove i predstavlja najveću sličnu organizaciju u Evropi. Izdaja časopis koji u poređenju se komercijelnim izgleda sasvim neugledno eli u njamu nema reklama e tekat iz koge se ne može naučiti ništa novo treba tražiti svećom. Što je najlepše časopis objavljuje programa koji se skoro mogu oceniti kao odlični i revni komercijelnima! Sa drugim sličnim klubom koji se zove Laserburg do sada nismo imali kontakt eli može da se pratpostevi de deluja na sličnim osnovema.

Sva u svemu BBC B predstavlja izvenreden izbor za onoga ko za računar može da potroši 400 funti. Traba zaista biti kritizarski raspoložen da mu se nadu mane i pored toga preporučujemo viesnicima BBC B de ulože još 100 funti i nabave proširenje od 20 Kb koje oslobada čitavih 32 Kb RAM-a bez obzira na grefički mod u kome se radi. Ovo proširenje pogotovu ako bude više programa koji bi koristili njegove usluge može da bude dobra zamena drugom procesoru!

Dejan Ristanović

commodore 64 računari u izlogu

Commodore 64 se pojavio na tržištu gotovo bez imalo galame, toliko uobičajena za promocije novih računara, ali je na njemu odmah stekao veoma čvrsta pozicija. Iako ja praksa iznela na videlo nekoliko njegovih dosta naprijatnih slaboeti, dobri znalci računareka tehnika smatraju ovaj računar jednim od najboljih modele u tzv. srednjoj klasi. Prava enaga ovog računara, ipek, dolazi do izražaja tak kada radi u sistamu sa evojim štampačem i avojom diak jedinicom i kada mu sa dokupa neka proširanja. Medu njih, na žaloet, spade i malo boiji — bajzik.

Poznavelac računara odmah zapaža nedostatak kablova za povezivanja sa kasatofonom COMMODORE 64 je konstruisan tako da traži avoj originalni kasatofon koma može sam da uključuja i iskljućuje motor Moguća je i povazivanja sa normalnim kasatofonima što traži malo poznavanja elektronika ali se tada moramo odrači programska kontrola rada motora

Kasatofon koji prodaja COMMODORE apaja se posabnim višežilnim kablom na računar Standardnog je kvaliteta sa brojačem Brzina komunikacija računara sa kasetofonom je vrlo mala, otprilika kao kod ZX—81 ali je pouzdanost izuzetna Svaki bajt se na traku anima dva puta i pri učitavanju se njihovim poređenjem konstatuje graška

„Rastegljiva" mamorija

Po uključenju se inicijalizuju boja slova akrana i okolina akrana redom na balu avetlo plavu i tamno plavu što se pokazalo kao vrlo iritirajuća kombinacija za oči pa je redovan postupak da se ta kombinacija promani recimo na zelenu, crvanu i crnu Kvalitet alika je vrlo dobar, bez uobičajanog pratifanja koja se javlja kod aoscirums

Na akranu se pojavljuja poruka 64K RAM SYSTEM 38911 BAJTA SLOBODNO Nija štamparska greška! Računar zaista ima 64K RAMA ali je za bejzik dostupno svega 38911 Razlog ovoma leži u koncepciji računara U njaga je ugrađeno 8K bajzik ROM-a 8K ROM-a u koma su rutina za rad sa diakom i printarom i 4K ROM u koma sa nalaze 4 karaktar sata po jedan za velika i mala slova i dva gratička sata

Na nakim mestima se ROM i RAM praklapaju Tada je ROM kao važniji jedino dostupan Za proaečnog korisnika dakle COMMODORE 64 ima pristupačno ralativno malo mamorije prama onoj količini koja je u njaga ugrađena Ostatak mamorija se može iskoristiti samo u mašinskim programima Tada se ROM može programski isključiti i tako priči memoriji koja se nalazi „pod njim

Uputstvo za programiranja je najslabiji deo ovog kompleta COMMODORE je poznat po lošim uputatvima koja ne sadrže ni deo potrabnih i neophodnih informacija



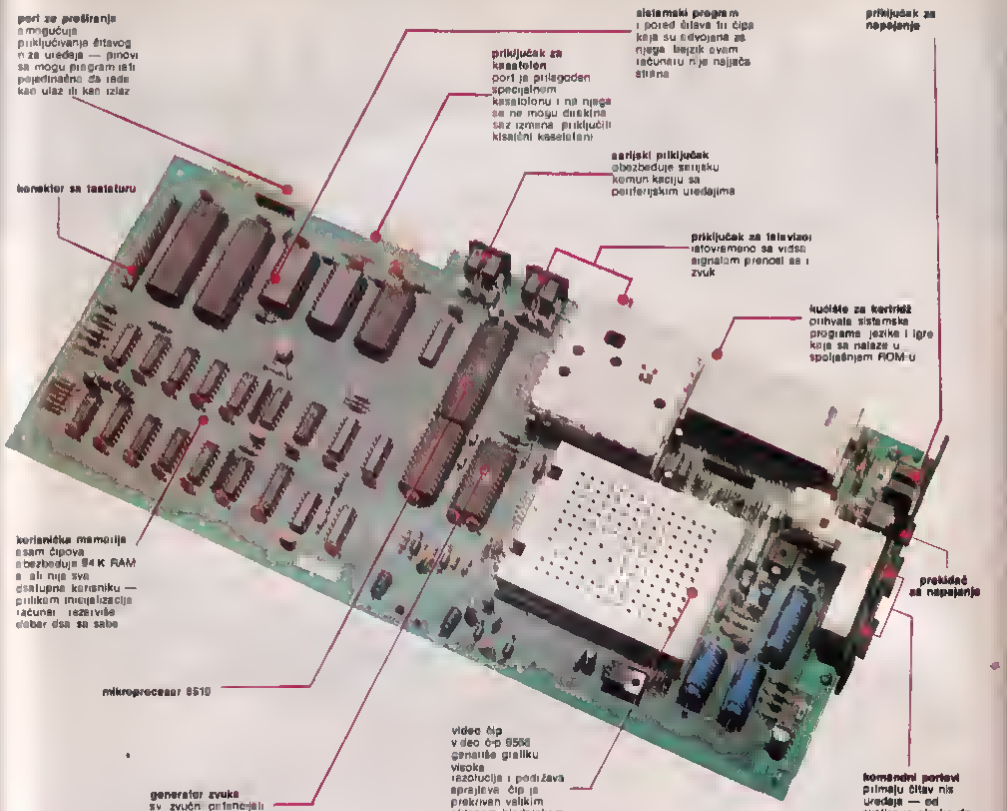
Nikoga ne ostavlja ravnodušnim Commodora 64

Nameće se poređenja sa Spectrumovim uputstvom koja je i po količini informacija i po kvalitatu daleko ispred COMMODOREovog Korisniku koji žali da programira neophodna je knjiga „COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE" u kojoj se nalazi nauporedivo više informacija o računaru Cena je 10 funti u Engleskoj

Sam računar izgleda impozantno sa pravom tastaturom od 66 tipki uključujući i izdvojena 4 funkcijska tastera sa dsana strana koji namaju prektično nikakvu namanu osim što se programski može registrovati pritisak ne njih Sa desna strana računara nalaze se dva konaktora za priključanje komandnih palica ili avetlosna olovke Na zadnjoj strani nalaze se konaktori za dodatni ROM TV priključak video izlaz Hi-Fi izlaz i ulaz (za ulaz nigde nama ni reči kako ga iskoristiti) za vezu sa diakom i printarom za kasatofon i takozvani „USER" port na koji se verovatno mogu priključiti naki dodatni uređaji i povazati paralalni printari

Naima veza računara sa diskom i printerom je sarijska što znači da se prenosi bit po bit informacija čima je izmena podataka izmadu računara i spoljnih uređaja ralativno apora Program od 20K, na primer, učitava se oko 50 sekundi sa diska U poređenju sa paralalnim diskovima to je prava večnoat No kako diak na sluš samo za animanja i učitavanja programa već i za rad sa podacima (RANDOM FILES na primer) kada se može vrlo brzo priči bilo kom podatku na diaku bez učitavanja prathodnih na traba ga se unapred odreći

COMMODORE 64 je bazirao na procesoru 6510 — nadgradnje poznatog procesora 6502 — iako 6510 ima nakoliko novih mogućnosti u odnosu na 6502 programiranja na njima je identično Ovi procesori su jednostavniji od poznatog Z80 i mašinako programiranja na njima je ralativno lakše Za displej se brina posaban čip koji se određenih lokacija u mamoriji uzima podatka i generiše aliku Postoja dva načina rada ovog čipa u niakoj i visokoj razolucija



Procesor: 6510
Klok: 1 MHz
Memorija: 64 K — 20 K ROM-a sa operativnim sistemom i bejzikom i 44 K RAM-a kada se ne koristi bejzik interpreter dostupno je maksimalno 54 K RAM-a
Ekran: 25 redova sa po četrdeset znakova visoka rezolucija sa 320×200 tačaka 16 boja
Interfejsi: kasetofon TV kertridž monitor 232 korisnički port
Jezici: bejzik
Jezici na raspolaganju: forti komal pilot logo USCD paskal i druge verzije bejzika
Tastatura: 62 mehaničke tastera standardne namene i 4 višenamenska
Dokumentacija: slaba nerazumljiva i nedovoljna
Dimenzije 404×216×75 mm
Cena: 223 funti

Grafika bez podrške

Niska rezolucija ili takozvani TEXT mod je „normalan" način rada Tada su na ekranu 25 linija sa 40 karaktera na svakoj U visokoj rezoluciji na ekranu su 320 linija sa 200 kolona Trebe odmah naglasiti da BASIC 2.0 koji je ugrađen u računar ne podržava visoku rezoluciju odnosno ne postoje komande za crtanje tačaka linija krugova i elipsa Za to će se potruditi SIMON'S BASIC koji se prodaje kao proširenje ROM-a ili na disku a u zadnje vreme može se naći i na kaseti po ceni od 190 DM Osim tih ovaj prošireni BASIC 2.0 ima još mnoštvo korisnih novih komendi između ostalog podržava muzičke mogućnosti koje je osnovni bejzik tekode zapostavio

Korišćenje visoke rezolucije i muzičkih pogodnosti moguće je doduše i na BASIC-u 2.0 ali isključivo POKE naredbom što je i dosta nepraktično i dosta sporo

Jedna od dobrih mogućnosti BASIC-a 2.0 je kontrole (opet POKE naredbom) tekozvanih sprejtova (SPRITE) Sprajt je karakter veličine 24 sa 21 tačkicu Kada se negde u memoriji nalaze 63 bajta koje određuju jedan sprajt dovoljno je postaviti jeden bit na 1 da bi se taj sprajt pojavio na ekranu Njegova X i Y koordinate se takođe nalaze u memoriji i mogu se promeniti POKE naredbom čime se sprajt celi sa jednog na drugo mesto brzinom kojom se izvršava POKE Svaki sprajt može biti u jednoj od 16 mogućih boja

U svakom trenutku se može kontrolisati do 8 sprajtova na ekranu Sprajtovi mogu biti povećani po X i/ili Y osi 2 puta pri čemu im se samo veličina udvostručuje dok rezulucije ostaje ista

Za svaki sprajt se može odrediti prioritet po kome se prikazuju kako za pozadinu tako i za ostale sprajtove Tako sprajt može stajati iza slova u TEXT modu a može biti i preko njih U slučaju kada se dva sprajta preklope delom ili potpuno sprajt sa višim

prioritetom će biti prikazan a onaj sa nižim će biti zeklonjen Video čip će se sam jeviti na pogoden način setovanjem odredenog bita ako je došlo do dodira dve sprajta što se može lepo iskoristiti u igrama

Sprajtovi su normalno u dve boje ali se eko se žrtvuje rezolucija po horizontali mogu pravliti i u četiri Gratika u visokoj razolucij je takode samo u dve boje ali i za nju vredi mogućnost 4 boja uz smanjenje rezolucije sa 200 na 160

Za muziku se brine posaban čip koji kao i video čip sa odredenih lokacija u memoriji uzima podatke za tonove Zalo je i moguće programiranje tih čipove običnom POKE komandom Mogućnosti zvučnog čipa su upravo izvanredne i mogu se meriti se mogućnostima skromnih sintesajzera Moguće je istovremeno korištenje tri zvučna kanale Na svakom od njih ze svaki ton se može odrediti njegov oblik pomoću 4 parametra (ADSR) e može se regulisati i jačina Povrh svega može se manjati i kvalitet tona odradivanjem telasnog oblika od koga se sastoji zvuk I na kraju zvuk se šalje kroz ententski priključak u televizor tako de mu kvelitet zavisi od kvalitete zvučnika u televizoru Hi-fi iziez je mono, eli se može jednostavnim elektronskim kotom elmuilrati i stereo

Bledunjavi bejzik

BASIC 2 0 predstavlja kiseličnu verijantu bejzika uz vilo melo odstupanja i zato se programer, koji je navikao na komotnosti koje pruža Sinklerov bejzik u Spectrumu, u njamu teško snalazi To se naročito odnosi na rad se slovnim verijablema, sa kojima se operacije više pomoću LEEFT $ MID $ i RIGHT $

Programe pisana na COMMODORE 64 je lakše prevesti na Sinklerov bajzik nego obratno Doste je nezgodne strane BASIC-a 2 0 što su nazivi varijabli ogiencni na svega dva karaktere i što se programska linije na može sastojati od više od 80 karaktera uključujući i broj linije Dobra strana mu je takozvani ekranski editor čime ja ispravljanje programskih linija znatno olakšano Ako se želi naka izmena u programskom listingu dovoljno je dati LIST komandu dotetati kursor na odgovarajuće mesto Izvršiti izmenu (obrisati zamenti ili umetnuti) i pritisnuti RETURN čima će u programski listing umesto stare biti unesena nova verijenta linije

COMMODORE 64 ima nekoliko nezgodnih osobina koje mogu često de zasmeteju Jedna je da brojevi programskih linija na koje skaču GOTO i GOSUB komande moraju obavezno da postoje u programu — i računar ne podrazumeve sledeću liniju eko nedostaje ona na koju se skače Druge osobina je da se sa svakim unosom bilo koje programske linije koje počinje brojem gube sadržaji svih verijabli Ta osobina je naročito naugodna pri testiranju programa primetile da program pogrešno redi izvršite izmenu (izmenita liniju i pritisnite RETURN) i ne možete nastaviti rad programa sa CONTINUE jer ste upravo izgubili sadržaje svih varijabli Jedino rešenje je ponovo RUN U slučaju greške na kraju dužeg programe (kome treba recimo 10 minuta da stigne do nje) biće vam jasna važnosi ove osobine

BASIC 2 0 dopušta ied se numaričkim verijablama čije ime mora zavišili znakom % Opseg brojeve se kreće izmedu — 32767 do 32767

Osim bejzika, postoje još neki programski jezici koji mogu iediti na COMMODORE 64 To su, pre svege, pilot, logo, tort, paskal i komal pri čemu su bez sumnje najznačajnije poslednja dve paskal je komi pajler sa vilo dobrim mogućnostime e komal je donekle sličan bajziku eli ima mnogo moćnije komande

Disk jedinica

Najpristupačniji (najjevtiniji) disk za COMMODORE 64 je VIC 1541 koji u Nemačkoj košta oko 700 DM Njegov kapacitet na jednoj disketi je 170K i do 144 fejle odnosno programa Za ozbiljno korišćenje računala neophodno je posedovanje bar jednog diska od četiri koliko se može istovremeno priktjučiti Kasetolon je jednostavno suvise spor Možda je za prosečnog korisnika dovoljno da nabavi progiem pomoću koga kaseloton snime i reprodukuje brzinom diska, eli ostaje činjenjca de to još nije disk u pogledu ostelih mogućnosti

VIC 1541 se sa računalom povezuje šestožilnim DIN kablom koji se dobije uz disk Na mrežu se povezuje svojim kablom Diskele koje se ulažu u disk su standardne veličine 5 25 inča (13 cm) Uz disk se dobije uputstvo koje je i ovog puta vrlo štuoi i sa malo neophodnih primera Uz to ima i dosta štamparskih grešaka i nedoteženosti što sve zajedno stvara loš utisak

Na demonstiecionoj disketi koje se takode dobija uz disk nalazi se desetak programa koji su samo donekle demonstracioni e više su korisni o kasnijem redu sa diskom i ze to COMMODORE treba pohvaliti

Njihovom analizom se može saznati mnogo onoga što nije rečeno u upuistvu Na disketi se nalazi i WEDGE koji služi de bi se izbegle dosle glomazne komanda koje su neophodne pri radu sa diskom DOS (DISK OPERATING SYSTEM) je ugreden u ROM računara i to predstavlja značajnu predmost u odnosu na druge sistema kod kojih se DOS nalazi na disketi i zeuzime njen veći deo a takode zeuzima i RAM u računaru kada se u njega učita

U disku se naleze procesor 6502 16K ROM i 2K RAM Dakle disk je dosla inteligentan Tehnički je izvodljivo spojno programiranje procesore u disku čima on preiazi na mašinski program koji se nalazi u ta 2K RAM-a

Matrični štampači

Od matričnih printera koji se mogu povezeti na COMMODORE 64 neophodno je pomanutr dve najbolja modela koji istovremeno imaju i dosta prislupačnu cenu

Prvi je VIC 1525 koji je u stvari prepravljeni SEIKOSHA GP100 u koji je ugraden DOMMODORE-ov intelljes On sa svojom cenom od oko 700 DM predstavlja vilo dobai izbor Poseduje nešto lošiji karaktei set kada su u pitenju mala slova ali je to sve vilo čittjivo Brzina štampanja mu je oko 50 karaktera u sekundi Šiiina papire koju može da prihvett je 10 inča što odgovera formatu A4 Papii more biti perforiran po ivicama Može se nabaviti i kod nas

Drugi je VIC 1526 koji verovatno, doiazi iz COMMODORE-ovog razvojnog biroe Cena mu je oko 900 DM Karaktei set mu je izvenieden e brzina štempenja zahvaljujući redu u oba prevca nešto veća nego kod 1525 Šiiina papire je ieta kao kod 1525 ali ne mora biti perforiren Može se štempati i na običnom A4 papiiu što predstavlje značajnu prednost

Obe ova printera rade sa 80 karaktera u redu i imaju mogućnost grafike SIMON S BASIC ima COPY komandu koja služi da sliku u visokoj rezolucij iskopira na printer U kombinaciji sa 1525 sve lepo radi e se 1526 dolazi do raspada sistema — printai i računar se zaglave pa se moraju isktjučiti oba COMMODORE je svojevremeno povukao iz prodaje celu seriju 1526 tei je greška u ROM-u koji se nelazi u printeiu Najveljena je nove serije sa istom oznakom 1526 kod koje će greška biti ispievljena Zato tiaba obavezno ispiobati printer 1526 sa SIMON S BASIC-om prilikom kupovina

Programska podrška

Programska podrške je doste solidna ali uglevnom preovleduju igie Pošto je COMMODORE 64 sa diskom i printeiom dosta pogodan za obiadu teksta ili neki manji biznis ima i takvih progiame Naravno oni cenom dosta odskaču od programa za igiu čije su cene od 5 do 15 tunli Za seda je najskupiji progiam SIMPLEX 64 koji je namenjen za unos i obiadu dnevnih bilansa neke manje firme sa cenom od 172 tunte

Postoji nekoliko progiema ze obiadu lekste WORDCRAFT 40 EASY SCRIPT VIZAWRITE Programi su profesionalno uieden i imaju odlične karaktaristike Teksi se može držati na disku izmena se mogu viši po želji itd

Sa cenom u Nemačkoj od oko 700 DM COMMODORE 64 je vilo pristupačan i našem džepu Ako se kupuje kod nas preko oglasa može se očekivati dupla ili veća cena DOMMODORE 64 ne preporučujem početniku ze njega je bolji Spectrum Za ozbiljnieg korisnika koji namerava da investira u disk i printer COMMODORE 64 je vilo dobai izbor Ako na želi da programira imaće podršku dobrih (i skupih) progiema Ako želi da piše svoje programe u toma neće daleko otiči bez SIMON S BASIC-a.

Ivan Gerenčir

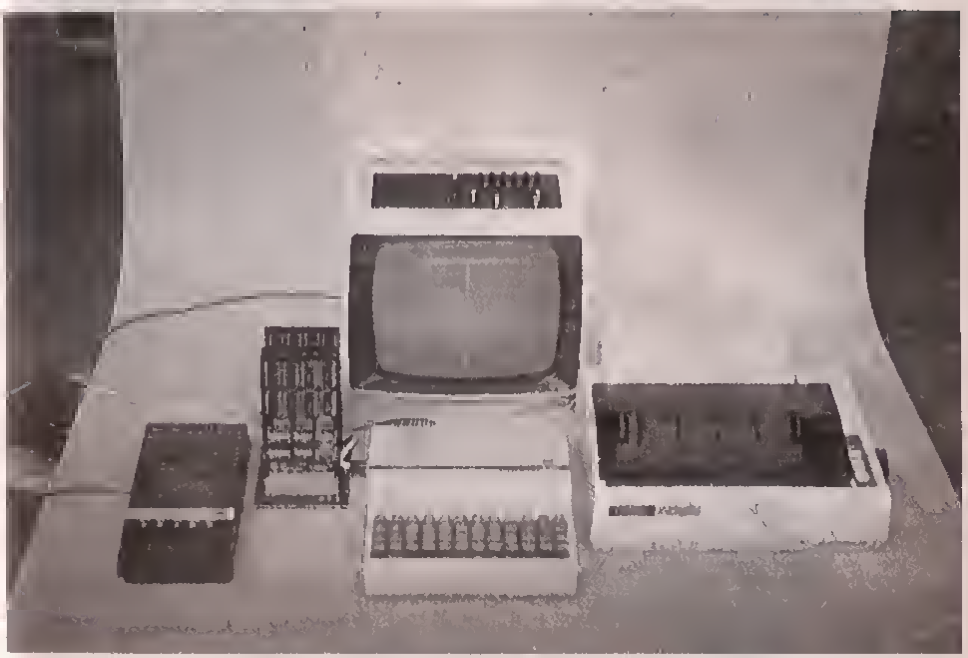
Naredbe i funkcije

ABS	INPUT	RESTORE
AND	INPUT#	RETURN
ASC	INT	RIGHT$
ATN	LEFT$	RND
CHR$	LEN	RUN
CLOSE	LIST	SAVE
CLR	LOG	SGN
CMD	MID$	SIN
CONT	NEW	SPC
COS	NEXT	SQR
DATA	NOT	STEP
DEF	ON	STOP
DIM	OPEN	STR$
END	OR	SYS
EXP	PEEK	TAB
FN	POKE	TAN
FOR	POS	THEN
FRE	PRINT	TO
GET	PRINT#	USR
GET#	READ	VAL
GOSUB	REM	VERIFY
GOTO		WAIT
IF		

IVO LOLA RIBAR

Industrija mašina
beograd
železnik
jugoslavija

složena organizacija
udruženog rada



U prvom izdanju Galaksije Računar u vašoj kući upoznali smo vas sa računarom **LOLA 8** i njegovim osnovnim mogućnostima U međuvremenu on se dalje razvijao i širio oblasti primena i krug ljubitelja

LOLA 8 je nastao u računarskoj laboratoriji LOLE RIBARA jer se pojavila potreba za malim računarom koji bi se jednostavno programirao a porad standardnih mogucnosti personalnih računara uz dodatne module koristio za obradu poslovnih i tehničko-tehnoloških informacija po potrebi i u realnom vremenu

BASIC je izabran kao najpristupačniji jezik za ovakvu vrstu računara Ipak je realizovan veoma moćan set instrukcija naročito u oblasti matematičkih operacija sa tačnošću do 10 decimala i velikom brzinom izvršenja

LOLA 8 je i bez posebne reklame stigao u mnoga škola fabrike institute projektne biroa laboratorije domove zdravlja kancelarije Veliko interesovanje i mogućnosti primene u raznim oblastima kao i iskusive u korišćenju računara **LOLA 8** su uslovili dalji rad na njegovom razvoju

Pokazalo se da su 6K reči RAM memorija nedovoljni za ozbiljnije programe na BASIC-u RAM memorija se proširuje modulima od po 16K Poređenja radi u 6K reči RAM memorije može da stane program od oko 150 linija (dužina 1 reda ekrana) 650 različitih promenljivih 100 vektora dimanzija 10 elemenata ili pak 230 nizova prosečna dužina 40 karaktera Dodavanjem samo jednog modula od 16K dužina programa može da bude i do 500 600 linija čak 2500 promenljivih, 330 vektora dimanzije 10 ili 950 nizova

Većina korisnika parsonalnih računara zna da je njihove slaba tačka nepouzdanost zapisa na kasalolonu koja zavisi od kvaliteta i tipa korišćenog kasalofona čisloče magnalna glava kasale I mnogih drugih faklora Rešenja spraga za kasaofon LOLE 8 omogućuje pouzdano snimanje I čitanje programa za gotovo sve kaselofone u komercijalnoj upotrebi Novost su I 3 nove BASIC komande

DSAVE DLOAD i DVERIFY koja omogućavaju snimanja I čitanje vradnosti promenljivih na kasetolon a vezana su za primena u postovnoj obradi informacija

Logičan nastavek razvoja računara **LOLA 8** je modul za priključenja štampača Programi pisani rukom su često nepregledni I naćilki pa se zbog mnogobrojnih izmena dešava da mislimo da smo unili jedno a u memoriji računara je nešto drugo Štampanjem sadržaja programa ovi problemi nestaju ali je još značajnija mogućnost štampanja razuliata obrada informacija

Većina štampača koja srećamo na inostranom tržištu ima paralelni ulaz/izlaz (tzv Cantronix standard) i svi oni mogu da se priključe na **LOLU 8** Situacija na domaćem tržištu štampača je dosta loše Postoji nekoliko domaćih proizvodaća (TERA TRS) ali su njihove cena za sada praviaoka

Zato je razvijen modul koji omogućave da se električna mašina za pisanje mogu koristiti umesto štampača Postoji nekoliko tipova ovih mašina na domaćem tržištu OLIMPIA ROBOTRON Ei Niš Njihova cena je znatno prislupačnija a veliki broj firmi ih već poseduje Naravno ova pisaće mašine ne mogu da pruže sve pogodnosti kao savremeni štampači ali su za većinu primena više nego dovoljne

LOLA 8, sa dodatnim modulima omogućava izgradnju malih sistema obrada informacija za različite primene evidencija knjigovodstvo ili praćenja i upravljanje jednostavnim procesima

Nadamo se da vas ovim poslednjim nismo preplašili Hlali smo da vam ukažemo na još neke mogućnosti primene **LOLE 8** U LOLI RIBARU svakog dana preko 30 učenika III razreda OC Beogradski bataljon radi na računaru **LOLA 8**, rađaju se ideje za nove igre svira Bahova svita u H molu drugim rečima stasava nova generacija koja zna šta je računar

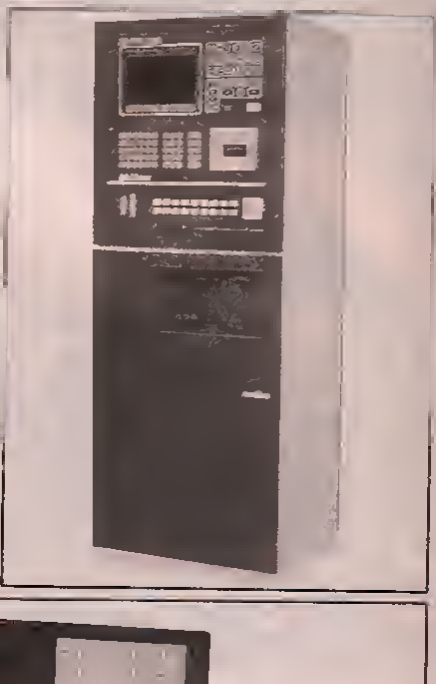
O saradnji LOLE sa srednjim i osnovnim školama I drugim mogućnostima računara **LOLA 8** recimo Ion generatora- u nekom od sledećih nastavaka Galaksyinog časopisa Računar u vašoj kući Dolle za sva obaveštenja obratite se na tel 570-227 ili pismom na adresu industrija mašina IVO LOLA RIBAR za računar **LOLA 8** 11250 BEOGRAD–ŽELEZNIK

LOLA RAČUNARSKI SISTEMI

Industrija IVO LOLA RIBAR jedan je od najznačajnijih jugoslovenskih proizvođača računarskih sistema namenjenih upravljanju u industriji. Familija LOLA računara izgrađena na bazi standardnih modula karakteriše se fleksibilnošću pouzdanošću u industrijskom okruženju i jednostavnošću eksploatacije koja ne zahteva posebne obrazovanje u oblasti računara.

Svi predstavnici LOLA familije industrijskih računara se mogu jeftino standardnog modula spregnuti sa centralnim računarom, omogućavajući lako izgrađivanje distributivnih sistema upravljanja za veće proizvodne linije i čitave fabrike.

Iskustvo više generacija stručnjaka u projektovanju i proizvodnji upravljačkih sistema i korišćenje najsavremenija mikroprocesorske tehnologije garancija su kvaliteta ILR proizvoda u oblasti industrijske elektronike. IVO LOLA RIBAR danas raspolaže timovima stručnjaka koji rade na razvoju novih računarskih sistema, obuhvatajući tako sve širinu potreba automatizacije u industriji.



LOLA SISTEMI NUMERIČKOG UPRAVLJANJA

U oblasti programskog upravljanja LOLA RIBAR predstavlja svoje sisteme numeričkog upravljanja LOLA 30 (za buduće glodalice obradne centre), LOLA 30S (za strugove) i LOLA 42 za upravljanja industrijskim robotima. Sva ti sistema se pravi do se upravljanje složenim kretanjima max. 8 istaknutih ili linearnih osa simultano.

LOLA 30 CNC (computer numerical control) obezbeđuje sve standardne funkcije sejmodernijih alatnih mašina

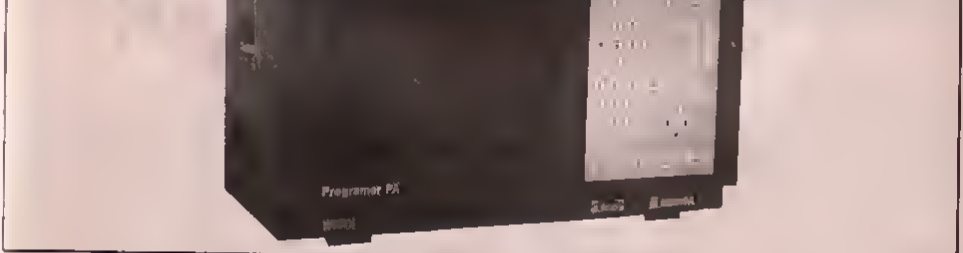
- Prilagođenje se različite tipove mašinskih sistema (se razlučivanjem 0.001 ili 0.01 mm)
- 3D linijsama interpolacija
- kružnom interpolacija (jedan krug) u tri ravni
- umetanje podataka preko čitača bušene trake, kasetofona, tastatura, perifernih jedinica ili alfanumeričke tastature
- Tabele korekture za max 30 alata
- Fiksni ciklusi bušenja i struganja
- Glodanje i rezanje navoja
- Apsolutno, inkrementalno i mešovito programiranje
- Parametarsko programiranje

Ekranske jedinice (31 cm) omogućava prikazivanje aktivnih kodova trenutnih pozicija svih osa aktivnog i delu sledeće bloka programa kao i svih potrebnih informacija, upozorenja i uputstava za operatora.

Integralni deo LOLA 30 sistema je programibilni automat PA 512 na bazi kog je moguće vana fleksibilna sprega sa mašinom.

Specifičnost LOLA 42 CNC (computer robot control) sistema za upravljanje industrijskim robotima je način unošenja programa u radnu obuzdavanja. Operator dovodi alat ili robota u željenu poziciju koja se automatski memoriše. Ovako unet program se zatim može izvršavati u automatskom pojedinačnom režimu.

Osim više linijske interpolacije kretanja upravljački sistem primenjuje i specijalne instrukcije simultanizacije rada robota sa tehnološkim okruženjem. U tom smislu na raspolaganju korisniku stoji 32 ulazno-izlazna signala koja se može povezuje sa drugim robotima ili mašinama.



LOLA PROGRAMABILNI AUTOMATI

Najveću promenu u svim granama privrede (eksploatacija i prerada uglja i gasa, petrohemija, velika industrija, transport, poljoprivreda i prehrambena industrija) je donela primena LOLA programabilnih automata. Pojava PROGRAMABILNIH AUTOMATA (skraćeno programabilni automati) se dimenzionisu prema konkretnom upravljačkom zadatku i mogu zajedno se njihovim eventualnim proširenjem.

Obezbeđuju realizaciju relejne logike sa novim savremenim tehnološkim, brojačke i vremenske funkcije kao i obradu numeričkih podataka. Za rešavanje problema regulacije u industriji razvijen je specijalan modul PID regulacije koji prema standardnim algoritmima obrađuje do 32 regulacione petlje.

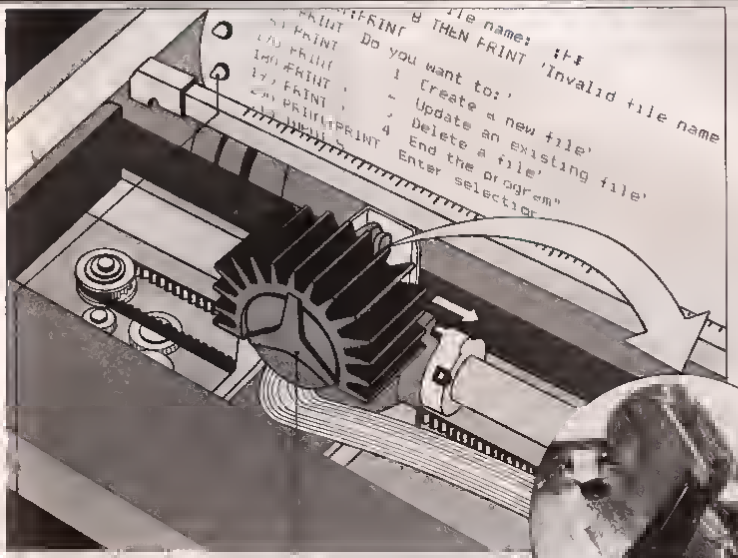
Poseban uređaj za razvoj i testiranje upravljačkih sistema na bazi programabilnih automata — PROG PA omogućava unošenje upravljačke logike u vidu standardnih relejnih šema uz istovremeno prikazivanje na ekranu. Unet program se može sačuvati u EPROM memorijama na magnetnoj kaseti ili sa priključenim štampačem može dobiti kompletna dokumentacija. Specijalni režim prikazivanje omogućava operatoru da u svakom trenutku prati se ekranu odvijanje procesa u realnom vremenu. Svi aktivni elementi relejnih šema prikazani su inverzno. Time se vrlo jednostavno da se uoči u kojoj grani u radu nastaje sveda na minimum.

LOLA programabilni automati (mini PA, PA 256, PA 512 i MPAJ koncipirani su tako da se zahtevanim ekonomskim efektom zadovolje širok opseg kolegijuma primene od minimalnih konfiguracija sa max. 32 ulazno/izlaza do složenih upravljačkih zadataka koji podrazumevaju na stotine velik broj ulazno/izlaznih signala (do nekoliko hiljada), veći visok nivo numeričkog procesiranja.

POSETITE NAŠE LABORATORIJE BUDITE SVEDOK RASTA MOĆNE LOLA ELEKTRONIKE I ROBOTIKE!

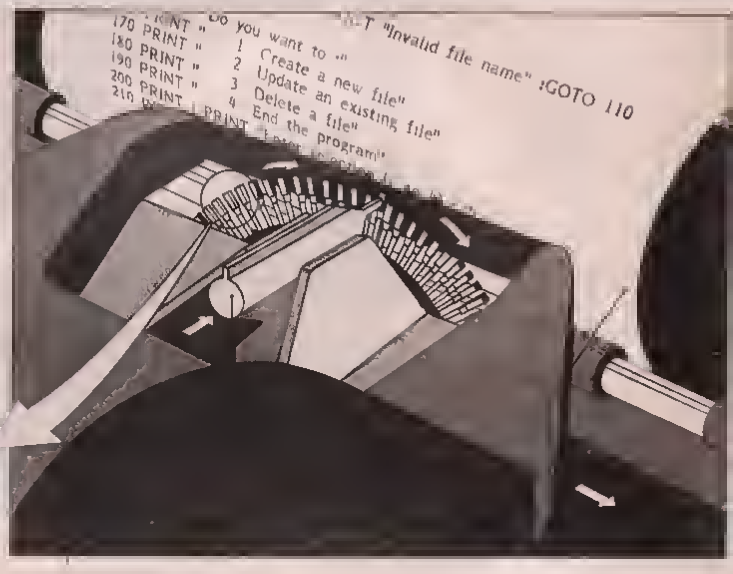
štampači u akciji

Periferijska oprema


PRINT
PRINT
PRINT

Štampač, bez sumnje, predstavlja najvazniju periferijsku opremu svakog kucnog računara koji svome vlasniku predostavlja nešto više od igre. Bez njega aa ne može zamisliti obrada teksta, rad sa datotekama, pe čak ni iole ozbiljnija programiranje. Na tržištu sa trenutno mogu naći dve lipa štampača — matični (dot-matrix) i sa lepezom (daisywheel). Prvi omogućuju definisanja posebnih karaktera (kao što su naša latinična slova) i crtanje i rada samostalno i veoma brzo. Drugi su previše spori, zahtevaju mahaničke intervencije na lepezi da bi sa uvele i neša slova i atalno opsluživanje papirom, ali imaju izvanredan otisak. Kvalitet otiska matičnih štampača ja dovoljan za ličnu upotrebu — listinzi, datoteka, dokumentacija, avidancije — ali na njamu na možete napisati tekst za novine, seminarski ili diplomski rad, pa čak ni posiovno pismo. Tu postoja samo dve mogućnosti: štampeć sa lepezom ili — ručni rad na klasičnoj pisacoj mašini.

Štampač sa lepezom
Lepezu na čijim se špicama nalazi po jedno slovo okreće veoma precizan motor, kada dode u poziciju za štampanje slovo otiskuje na hartiju elektromegnatski udareč, otiskuje slovo na hartiju mašina i štampača sa lepezom nema značajnijih razlika u kvalitetu otiska... ali je rad sa štampačem neuporedivo udobniji jer ja pra štampanja moguće sve ispravke na tekstu izvršili efektronskim putem





okviru jednog reda možemo slobodno da mešamo obične dvostruke i karaktere sačinjene od kolona koje smo definisali bez rkakvih ograničenja što nam omogućave pisanje normalnog latiničnog tekste pa i grčkih slova potrebnih za neko neučno štivo

Seikosha GP100 ne spada u brze štampače — ispisuje samo 50 karaktera u sekundi što bi bilo sasvim dovoljno da je štampač -pameten : da piše i dok se glava vraća u početni položaj To na žalost nije slučaj ps ispisivanje jedna strana tekste ume da potraje prilično dugo Za večinu primena ovo nije velika smetnja Posebno ograničenje u brzini predstavlja okretanje doboše koje je sporije nego kod mnogih drugih printera — prelazak na sledeću stranu ume dakle i da potraje GP100A je uz to i pritično bučan štampač što neće biti previše omiljeno u vašem domu

Autoru ovoge tekata na smeta ni jedno od ovih ograničenje jer mu je jasno da je ono rezultat težnje da štampač bude jaftin Pri izradi sottvera koji je smešten u ROM se medutim škrtarilo daleko više nego što je dopustivo GP100A je teko ostao veoma -neinteligenten uredaj ne zna kada nalazi kraj strane pa ća pisati po perforaciji ukoliko se programer ne pobrina da prebroji ispisane redove ne omogućava softversko smanjenje broja slova u redu niti promenu razmaka izmedu redova pravi neobično mnogo problema kada programer poželi da piše dve puta preko istog teksta (radi podvlačenja veżnih poruka na primer) i što je posebno zamorno ima neprijetno dugo pamćenje ako u jednom trenutku nesmotrenosti pošaljemo nekoliko karaktera u njegov befer ne postoji regularan način da ih odatle obrišemo pra nego što red bude štampan (isključivanje štampača iz napajanja jasno ne smatramo regularnim rešenjem jer uvek može da rzazove probleme kod drugih istovremeno priključenih uredaja) Ne bi bio nikakav problem da je firma upotrebila neki od ionako neiskorišćenih kontrolnih karaktera za reinicijalizovanje štampača ovakav propust jednostavno navodi na pomisao de firma nije želela da ugrozr prodaju svojih daleko skupljih modela i da je zato GP100A ostao bez bolieg ROM-a

Uz štampač se dobrja knjižica od 30 strana kao jedina dokumentacija Za iskusnog korisnika računara dokumentacija je sasvim dovoljna, a obuhvata čak i sve što je potrebno za gradnju interfejsa za lične potrebe Za početnike ja knjižica medutim previše iskonski pisana čini nam se da će neko koma je GP100A prvi štampač utrošiti časova, a možda i dana da shvati ono što je mogio da buda jednostavno objašnjeno Žalosna je činjenica da oni koji pišu uputstvo za korišćenje računera izbegavaju da se pozabave radom sa štampačima (svakako zato što ima mnogo nekompatibilnih tipova) računajući da će ovaj najpopularniji periferijski uredaj biti posebno dokumentovan, a oni koji pišu uputstva za upotrebu štampača računaju sa tim da je onome ko poseduja računar dovoljna dokumentacija koju već ima Malo više standardizacije i saradnje poštedelo bi početnike mnogih glevobolja

Uve u svemu Seikosha GP100A ja umereno dobar izbor i verovatno može da se pohveli najboljim odnosom mogućnosti/cene Začudo nije postigeo prevelilku popularnost (možda će uskoro biti bolje?) što se odrazilo na slabiju softversku podršku Sof-

Privlačan za hobiste lako nema baš najbolji kvalitet otiska štampač Seikosha 100 ipak u odnosu na cenu predstavlja veoma dobar izbor

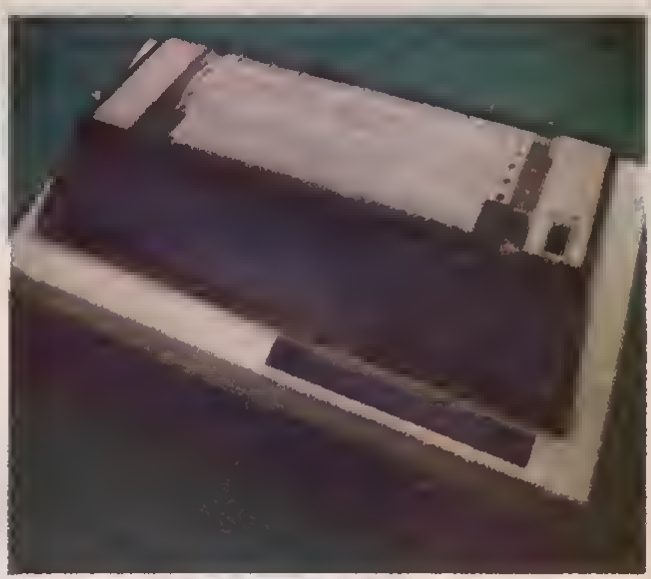
tverska podrška za štampač? I ona ponekada može da bude važna ako imate BBC B ili Electron na tržištu i po časopisima ćete naići na gomilu programa koji omogućavaju obradu teksta na Epsonu kopiranje sadržaja ekrana na Epsonu stvaranje utiska o bojama na Epsonu i mnogo što-šta slično Ukoliko na nameravate da sami pišete ovakve programe (nije teško ali oduzima vreme) zaboravite na Seikoshu i počnite da razmišljate o Epsonu kome ćemo i mi naravno posvetiti sledeća poglavlje

Epson FX80 i RX80

Firma Epson danas drži vodeće mesto u proizvodnji i prodaji štampača za kućne računare Za to može da zahvali prvenstveno svoma modelu MX80 i drugim štampačima iz MX serije koji su postavili standarde za mnoga druga proizvođača No vreme čini svoje pa su štampači iz serija MX postali preskupi Epson je odlučio da pripremi dovoljno jeftin štampač koji će dostojno *zaprezentovati* renoma firma I nastali su FX80 i RX80 FX80, zapravo ima ava dobra osobina štampača iz MX serija a košta samo 380 funti (320 bez interfejsa) što je saavim dovoljno da vem ga najtoplije preporučimo

FX80 je, naravno, matrični štampač koji formira slova na matrici 5×8 što znači da slova g, j i slična ne bivaju datormisana kao kod Seikosha Set karaktera je više nego univerzatan r obuhvata preko 250 simbola Odakle tolika slova? FX80 osim standardnih ASCII *karaktera* ima i nekoliko vrsta slova, kao i karaktere od interesa za razne jezika (naših slova na žalost ni ovde nema) Što se vrsta slova tiče na raspolaganju au normalna (80 u redu) povećana (40 u redu), kondanzovana (137) kondenzovana-povećana (68) slitna (96) i elita-povećana (48 slova u redu) Sve ova kategorija slova au uglavnom objašnjena samim imanom osim jednih — slitnih Elitna slova bi trebala da se što više približa kvalitatu koji se postiže na pisaćoj mašini odnosno da ulepšaju izgled dokumenta Zašto bi nako onda koristio bilo koja druga slova? Jednostavno zato što mnogi vola da dokument lapisan na štampaču izgleda u pravom smislu kompjuteraki

Epson FX80 je jedan od vrlo retkih štampača koji omogućavaju automatsko podvlačenja teksta i takozvani double strike mod O čemu se radi? Ponekad nam je potrabno da posebno istaknemo deo teksta ili jednostavno dobijamo kontrastniji dokument koji će ići u štampu Tada čemo izabrati opciju koja štampaču nalaže da svaki red štampa po dva puta pri čemu će drugi trag biti za pola tačka pomaren u odnosu na prvi (ava tabele u Računarima 1 au na primer ovako kucana) U tom slučaju štampač zbog povećana preciznosti ne štampa u dva smara što značajno umanjuje brzinu koja u normalnim usiovima doatiže više nego dovoljnih 160 karaktera u sakundi Postoji još jedan način da sa razlogom prepolovimo ovu brzinu ukoliko se odluči-



Seikosha GP 100A

Tip matrični
Matrica 5×7
Interfejs centronics
Memorija 256 znakova
Linija po inču 5
Karaktera po liniji 132
Brzina štampanja 50 znakova u sekundi
Karakter u redu 80
Broj kopija 1
Podvlačenje ne
Isticanje da
Proporcionalni razmak ne
Blok grafika ne
Grafika visoke rezolucije da
Dvosmerno štampanje ne
Logično traženje ne
Maksimalna širina papira 10
Pomeranje papira traktor
Dimenzije 42 7×23.0 cm
Masa 4 5 kg
Cena 247 funti



„Kraljavska loza" među matričnim štampačima Epson MX 80 može da zadovolji potrabe i najrazmaženijeg ljubitelja kućnog računara ali je za njega potreban više nego dubok džep

Epson FX 80

Tip: matrični
Matrica: 9×9
Interfejs: centronics
Memorija: 256 karaktera
Linija po inču: 6.8
Karakter po inču: 5, 8.2, 5, 10, 1", 5
Brzina štampanja: 80 znakova u sekundi
Karakter u redu: maksimalno 132
Broj kopija: 3
Podvlačenje: da
Isticanje: da
Proporcionalni razmak: opciono
Blok grafika: ne
Grafika visoke rezolucije: da
Dvosmerno štampanje: da
Logično traženje: da
Maksimalna širina papira: 25 cm
Pomeranje papira: traktor
Dimenzije: 40.6×38.1 cm
Masa: 5.5 kg
Cena: 400 funti

mo za 60 karaktera u sekundi štampač će pisati daleko tiše što je korisno ako radimo u nekoj kancelariji ili jednostavno ako nam ukućani ne tipe buku.

Za one koji su navikli na skupe pisaće mašine kod kojih su slova proporcionalna razmaknuta (m na primer ne zauzima onoliko mesta koliko i i) FX80 nudi opciju koja omogućava čak i takvu pogodnost. Da stvar bude još lepša bilo koja druga opcija (npr. veličina slova) može da bude izabrana zajedno sa proporcionalnim razdvajanjem slova.

Onima za koje ni ovoliko slova nije dovoljno na raspolaganju su naredbe za definisanje karaktera. Možemo na primer da predefinišemo neki nepotreban ASCII karakter i dobijemo naše slovo Č koje će samim tim biti dodeljeno nekom tasteru (ako imamo BBC ili Electron slekat ćemo kompletirati predefinisanjem tog karaktera u setu slova računara pa ćemo na ekranu videti baš ono što treba da se pojavi na štampaču). Interesantno je medutim zapaziti još nešto posle malo eksperimentisanja sa definisanjem karaktera primetićemo da naša slova izgledaju ružnije od originalnih. Da li to znači da imamo manje dizajnerskog talenta? Možda ali se i Epson pobrinuo da tako bude: kod slova iz ROM-a se ne adresiraju samo tačke već i prostori između njih (po pola tačke) što jasno poboljšava izgled dokumenta. Ta mogućnost na žalost nije ostavljena i korisnicima RX80 ne omogućava definisanje karaktera.

Epson FX80 je izvanredan za one koji žele da citaju. Moguće je naravno kontrolisati svaku tačku u redu ali koliko ima takvih tačaka? Skoro da bismo mogli da kažemo koliko poželite u normalnoj rezoluciji na svakih 6 inča (20 cm) ima 480 tačaka u dvostrukoj rezoluciji na taj prostor stane dvostruko više (960) ali je na raspolaganju i takozvana četvorostruka tačnost koja nudi 1920 tačaka! Ovakva rezolucija se približava ploterima i omogućava na primer generisanje bar koda koji bi razumeo HP41C ili neki drugi računar opremljen optičkim čitačem. Osim ovih mogućnosti na raspolaganju su i CRT grafika i II (640 odnosno 720 tačaka na 20 cm) koje su idealne za vlasnike BBC računara jer se podudaraju sa maksimalnom rezolucijom na ekranu računara. Za šta oni mogu da koriste veće rezolucije? Vešti programeri su već sastavili rutine koje sanče sirka čineći tako da se stekna utisak o bojama koje su bile zastupljene na ekranu koji je dampovan.

Za crtanje je vrlo korisno što može da se izabare razmak između redova i to sa velikom preciznošću (od šestine do dvesta šezdesetine inča). Može osim toga tako da se izabare položaj leve i desne margine (samim tim i broj slova u redu) pomeranje početka teksta od vrha i dna stiana i mnogo što-šta drugo. Vidimo da za razliku od Seikosha Epson FX80 prepoznaje kraj strana ali on radi i više od toga prepoznaje i zvučno signalizira trenutak kada je vaša zaliha papira pri kraju. U istom trenutku štampač koči kompjuter i čeka da stavite još papira. Jednostavan pritisak na taster ON LINE će zatim naložiti računaru da nastavi sa slanjem karaktera. Osim tastera ON LINE na kućištu printera su i tasteri LF (pomeranje papira za jedan red) i FF (pomeranje na početak sledeće strana) koji rmaju i sekundarnu ulogu: ako pritisnete LF dok uključujete štampač biće obavljena automatska dijagnostika (korisno kada kupujete štampač) a ako u tom trenutku držite pritisnute FF i LF svi podaci koje kompjuter bude slao biće štampani kao heksadekadne vrednosti (čak i kontrolni karakteri).

Osim tastera na kutiji na samoj štampanoj ploči printera postoji četvora serija mikroprekidača. Uz njihovu pomoć možemo da biramo stanje u kome će naš štampač biti kada ga uključimo. Ako na primer stalno koristimo stotine slova nema nikakvog smisla da posle svakog uključivanja šaljemo četvoru seriju kontrolnih karaktera — jednostavno ćemo pomeriti jedan preklopnik. Slično se odnosi i na druge opcije koje se u toku rada softverski menjaju kao i na setove slova koja biramo (RAM u kome se nalaze karakteri koje smo mi definisali biva jasno izbrisan kad god isključimo štampač).

Kvalitet je vrlo dobar. Ako želimo da štampamo nešto sa velikom brzinom onda je to moguće. Ako želimo da štampamo nešto sa velikom brzinom onda je to moguće. Ako želimo da štampamo nešto sa velikom brzinom onda je to moguće.

Epson FX80 u standardnoj varijanti ima Centronics paralelni interfejs ali se za njega može dokupiti i RS232. Obzirom da baler ovog štampača ima čitava dva kilobajta. Centronics interfejs će omogućiti brži prenos i manji gubitak računarevog vremena na beskorisno čekanje. Ovaj bafer na žalost ne može da se koristi ukoliko definišemo nake karaktere (RAM je ograničen) što znači da je za većinu primena u kojima su potrebna slova Č Ć Š i računar čekati da bude ispisano svako slovo pa će i RS232 biti više nego dovoljan.

Treba stvarno biti veliki probirač da bi štampaču FX80 uputile neka zamerka. Da pokušamo ipak i to Pisanje uputstvo je prilično opširno ali ne i naročito napisano nedostaju upravo one informacije koje ljubitelju računara koji je kupio svoj prvi štampač bile od neke koristi. Teškoćama snalaženju potpomaže i nekoliko bagova u operativnom sistemu štampača neki kodovi u nekim slučajevima (kada smo na primer odlučili da pišemo nekim posebnim slovima) jednostavno na iade ono što bi trebalo da iade dok drugi izazivaju neke neprijatne afekte. Moguće je medutim da su ti bagovi već otklonjeni ili da će uskoro biti.

Sva u svemu Epson FX80 predstavlja izvanredan izbor kako za one kojima je potreban biz printeri sa velikom grafičkom rezolucijom tako i za one koji zele kvalitetan izlazni dokument. Kod njega medutim treba obratiti pažnju na jednu stvar štampač je namenjen novim računarima kod kojih je set naredbi koje podržavaju grafiku veoma moćan. Starijim računarima (TRS 80 na primer) i novim računarima sa skromnom grafikom (mislimo naravno na računar galaksija) može da zasmeta činjenica da FX80 ne podržava blok grafiku što ograničava direktno prenošenje sadržaja ekrana na štampač. Blok-grafika se jasno može zameniti definisanim karakterima ali takvo rešenje zahteva učitavanje posebnog programa u računar posla svakog uključivanja štampače. Osim toga njegov ROM od 16 Kb je kompaktan (EPROM 27128) pa nije lako pronaći način da se u njega upiše neka naša slova. Oba ova nedostatka ostaju bodena je treći štampač o kome ćemo govoriti. Okidata Microline 80. Microline 80 je stariji sporiji i dalako slabiji štampač od FX80 ali kao što ćemo videti predstavlja prihvatljivu alternativu za mnoge primene.

Oki Microline 80

Microline 80 je da to kažemo na samom početku prilično stari štampač što u kompjuterskom svetu svakako nije naročita preporuka iako se jasno radi o štampaču opšte namene koji može da bude opremljen raznim od standardnih interfejsa. Microline 80 (u daljem tekstu jedino radi kraćeg pisanja MC80) specijalno je pripreman za računar TRS 80 (odatle verovatno oznaka). Za potencijalne vlasnike štampača u Jugoslaviji MC80 ima nekoliko velikih pluseva — piše pre svega na običnom računarskom papiru (oba kao što smo rekli mogu da se nabave kod nas) a ne uvek jednostavnije tražiti standardni papir) koristi običnu traku za pisaću mašinu i najzad njegova blok grafika je kao stvorena za naš računar galaksija.

Set karaktera MC80 je više nego kompletan pored velikih i malih slova brojeva i standardnih ASCII specijalnih simbola tu

Oki Microline 80

Tip: matrični
Matrica: 5×7
Interfejs: centronics
Memorija: 256 znakova
Linija po inču: 6.8
Karakter po inču: 5, 10, 16.5
Brzina štampanja: 80 znakova u sekundi
Karakter u redu: maksimalno 132
Broj kopija: 1
Podvlačenje: ne
Isticanje: ne
Proporcionalni razmak: ne
Blok grafika: da
Grafika visoke rezolucije: ne
Dvosmerno štampanje: ne
Logično traženje: ne
Maksimalna širina papira: 9.5 inča
Pomeranje papira: valjak, traktor, frikciono
Dimenzije: 34.2×24.5 cm
Masa: 6.5 kg
Cena: 242 funte

Izvanredan kvalitet otiske. Štampač sa lepezom Juki 6100 predstavlja jedan od najboljih modela u svojoj klasi

su 64 simbole od kojih je neke vrlo teško protumačiti (gledajući ih autor ovoga teksta dolazi na smešnu pomisao da su dizajneri želeli de omoguće igru „vešanje" (hengman) pa su se potrudili da pripreme karaktere u vidu čovečuljka kome nedostaje pola tela samo ruka i kso i karakter koji bi lepo izigrevao scenu as kraja igre) Niko se nerevno neće buniti zato što mu je set karaktere povećan eli vlasnici MC80 za razliku od vlasnika GP100A i FX80 nemaju mogućnost de se dopune sopstvenim karakterima što je ozbiljan hendikep šta ako su nam potrebne naša latinična slova? Odgovor na ovo pitanje možemo da damo samo onima koji poseduju programetor EPROM-a dovoljno je de izmenite deo štampačevog generatora karaktere, kodiranje je vrlo jednostavno praktično kso kod računara galaksije' (pogledajte „Računare 1" i tekst „Prvih nekoliko mikrosekundi")

Pored kompletnog seta karaktere MC80 ima i nekoliko veličine slova u redu možemo de štampamo 132 80 ili 40 karaktele (takozvane „duge linije") odnosno 105 64 ili 32 ksrektera („kratke linije") Da li je ovolika raznovrsnost potrebna? Možde i nije all su na ovaj način pomireni mnogi standardi štampenje 132 ili 80 karaktera u redu kopiranje TRS-ovog ekrena (64 slove u redu) i nerevno listenje galeksijinih Spectrumovih itd programa (u doba kada je MC80 projektovan bilo je raletivno mato računare koji imaju 32 karaktere u redu eli se delekovidost ttrme Oki danas pokazuje u punom sjaju) Brztna pisanje relko prelazi 50 karaktera u sekundt (zavisi od broja slove u redu i razmaks redova i može da bude najviše 80 CPS) što je umareno sporo eli prihvatljivo

MC80 nema gratiku visoks rezolucije što znači da korisnik jedino može da kontroliše grefičke blokove koji su veliki koliko šestina svekog karaktere. On je dakle više nego nepogoden za vlesnike BBC-je Electrona Commodore 64 i za one vlasniks Spectruma kojima štampeč treba da služi za kopirenje slika sa ekrana

I pored toga što mu nedostaju kodovi za kontrolu gretike visoke rezolucije, MC80 ima više kontrotnih kodova od Seikosha GP100A koji je u ovom prikazu već okarakterisan keo neinteligenten štampač' Znači li to de je MC80 „pametniji"? Izgleda da jeste iako ni on ne može da prepozna kraj strana (zar je toliko teško smestiti u RAM štampača jeden brojač redova?). MC80 nepogrešivo detektuje tranutak kada je list običnog papire prt kraju (tada za njim na sledi još jedan tist) kao i tranutak kada rolna kompjuterskog papire spadne na malu debljinu Osim toga ne njemu se nalazi taster pomoću koga štempač prelazi u mod OFF LINE tj ksde kompjuter more da čeka na našu intervenciju de bi nastevio de štampa Takve mogućnost nedostaje Seikosi — kada je baš neophodno da štampač „zakoči" računar pribegava se njegovom isključivanju čime se jasno gubi eve iz internog batere

Uz MC80 se dobije dobro uputstvo na 56 strana, u njemu se relevantne činjenice



Juki 6100

Tip lepeze
Matrica —
Interfejs centronics
Memorija 2000 znakova
Linija po inču 6 8
Karakter po inču 10, 12 15
Brzina štampenja 16 karaktera
Karakter u redu maksimalno 165
Broj kopije 4
Podvlačenje de
Isticanje de
Proporcionalni razmak da
Blok gratika ne
Gratika visoke rezotucije ne
Dvosmerno štempanje de
Logično traženje de
Maksimalna širina papire
Pomaranje papire trikciono
Dimenzije 454×520×151
Masa 14 kg
Cena 400 tunti

ponavljaju i po nekoliko puta all je to po nešem mišljenju velika pogodnost za početnike kojima je često teško de prete terminologiju pogotovu kede je tekat pisan na stranom (engleskom) jeziku

Sva u svemu sleki smo utisak de je Microtine 80 dobar štampač eli to ne znači de vam ga pretoplo preporučujemo njegova osna od oko 270 tunti nije toliko manja od Epsonovih 320 tunti (oba osne bez uračunatog interfejsa) da bi se čovek opredelio ze ipak deleko slabiji uredaj. Došlo je vrema računara sa visokom grafičkom rezolucijom i šteta je de tu razoluciju ne prati i printer!

Štampači sa lepezom

Štampači sa lepezom (daisywheel) do skora su bili preskupa alternativa brzim i pogodnim matričnim modelima U postednje vreme su razvijeni jettiniji margarita štampači (popularni naziv za štampače koji proizvode tekst okretanjem specijalna lepeze sa slovima) koji daju izvenredan izlazni dokument botji nego na elektničnot pisaćoj mašini Ovi štampači danas koštaju svega 300—500 tunti što nije mnogo viša cena od one koju plećamo za neki matrični štampač Pitanje jej dakle za kakav se printer odlučtil?

Pre nego što pokušamo de odgovorimo ne njega prikazaćemo bar ukratko, nekoliko štampača sa lepezom Brother HR15 Silver Reed EX44 Juki 6100 i Dateline 1200 DL

Pomalo je paradoksalno reći da najbolji kvalitet pisanog teksta daje najjeftiniji od pobrojanih štampača — Silver Reed EX44 koji košte 300 tunti Za ovaj novac se zapravo dobija pisaća mašina koje je opremljena interfejsom za povezivenje sa računerom što je idealno za ona koji se lek priviksvaju na korišćenje tekst-procesora

EX44 je na žalost prilično spor štempač — njegovih 12 karaktere u sekundi postiže i prosečan dektilograt Ovako mate brzina je rezultet kompromisa mogućnosti/cena EX44 ima bafer od samo 15 bajta štampa samo u jednom smeru i nema mogućnosti logičkog traženja Sve u svemu odličan štempač za one kojima se mnogo ne žuri!

Juki 6100 kao i mnogi dobri i jeftini uredaji dolazi iz Japana Njegove osna kade je opremljen interfejsom i potrebnim kablovima nešto je manja od 400 tunti Štempač je brži od prethodnog (16 znakova u sekundi) što je dopunjeno mogućnošću dvosmernog štempanje i internim baterom od 2 kilobajta Osim ispisivenje normalnih slova Juki 6100 omogućava proporcionalno razmicanje slove i slične pogodnosti koje imaju deleko skupiji štampeči Na žalost mehanizam koji okreće traku nije najsrećnije rešen pa se mastijeva traka ponekad zamoteva oko njega Druge loša strena Jukija je što je neprijatno bučan što se tek delimično rešava poklopcem koji se uz njega dobija

Čini nam se de je najbolji izbor Datalina 1200 DL Očekuje se da će njegova cena biti znatno povoljnija (sada 450 tunti) što bi,

U pripremi
RAČUNARI 3
u vašoj kući

- **NAKERSKI MANIFEST**
  Šta piše u jednom papiru koji tajno kruži među programarima sveta — šta jednom programeru dolikuje a šta ne i keko prepoznati pravog programara
- **RAČUNARI U IZLOGU**
  novitet! na evropskom tržištu — šta mogu, koliko koštaju a koliko vrede, kako se nebevijaju
- **RAČUNARI U AKCIJI — DISK-JEDINICE**
  disk jedinice su postale dovoljno jeftine i za kućnu primenu — kako izbor primariti svojim potrebama i mogućnostima računera — karakteristike kvalitet cene izbor nebavka
- **INTERFEJSI**
  izbor i povezivanje interfejsa za štampač sa računarom ponekad nije nimalo jednostavna stvar — kako se pišu softverski interfejsi — prikaz svih interfejsa za računar Spectrum
- **NEKE NOVE IGRE**
  prikaz najnovijih igara za računare Spectrum Commodore 64 i Electron
- **SOFTVERSKA TRPEZA**
  uporedni tabelarni prikaz sistemskog i uslužnog softvera za računara Spectrum Commodora 64 Electron i BBC B
- **SPEKTROTEKE**
  kako koristiti kućni računar za obradu podataka — prikaz svih datoteka za Spectrum — koja daloteka u kojoj prilici
- **ŠKOLA AVANTURISTIČKIH IGARA**
  kako se igraju a kako pišu igre-avanture
- **OPERATIVNI SISTEM/ADRESNI DEKODERI**
  prvi nastavak softversko-hardverske škole za buduće graditelje računara koja vas korak po korak praktično uvodi u hardversku i programsku koncepciju kućnog kompjutera
- **MAŠINAC ZA POČETNIKE**
  paralelna škole mašinskog programiranje u stripu za procesora Z80 i 6502 sa ilustrovanim rečnikom mašmonika
- **NAŠI PROJEKTI**
  — Irotonski generetor zvuka za računer „galaksija"
  — uređaj za automatsko učitavanja EPROM-a prilikom uključivanja računara ili nakon hardverskog reseta
- **COMMODORE 64**
  specijalan prilog za vlasnika računara Commodore 64 koji se i kod nas polako probija ke vrhu rang-liste najpopularnijih računara
- **BIBLIOTEKA PROGRAMA**
  novi programi za računara „galaksija" Spectrum, Commodora 64 i BBC/Electron

... NIZ NOVIH TEKSTOVA O RAČUNARIMA „GALAKSIJA" I SPECTRUM

Ukoliko želite da već sada obezbedite svoj primerak Računara 3 možete naručiti na adresu „Galaksija", 11000 Beograd Bulevar vojvode Mišića 17 — isporuka pouzećem

kombinovano sa dobrim karakteristikama ovoga štampača, moglo da mu pomogne da postane bestseler. Maksimalna brzina štampanja je pristojnih 25 karaktera u sekundi, pri čemu štampač piše u oba smera sa mogućnošću logičkog traženja. Bafer od 512 bajtova i Centronics interfejs omogućavaju povezivanje sa većinom postojećih računara na štampanoj ploči se nalazi određani broj prekidača koji omogućavaju izbor brzine prenosa.

Čak i onima koji plenirraju nešto da crtaju 1200 DL može da posluži umesto matričnih štampača njegova rezolucija od 165 karaktera na 11 inča (oko 28 cm) i minimalni razmak linija od samo 1/48 inča (0,021 cm) nisu karakteristike koje se lako mogu zanemariti.

Pre nego što se odlučite na kupovinu štampača sa lepezom morate ozbiljno da razmislite o jednom problemu: retko ko koristi štampač samo za pisanje teksta na stranom jeziku. Što znači da printer bez slova Č Ć Š i Ž nema mnogo smisla. Kod ovih štampača slova ne mogu da se definišu, što znači da je potrebna jugoslovenska lepeza koju koliko znamo niko ne može da vam isporuči. Jedan servis za precizmu mahaniku u Beogradu je krača vreme pokušavao da prepravlja lepeze za naše usisove eli su rezultati bili nezadovoljavajući. Neprijatno, zar ne?

Zamislite, osim toga, da ste napisali neki tekst od slotinak strana uredili ga pomoću računara i poželeli da se ispišete na štampaču. Štampači sa lepezom, jasno, pišu na običnoj hartiji, što znači da će vam biti potrebno nekoliko časova sedenja pored opreme i menjanje listova svaka dva minuta! Ništa, međutim, nije savršeno. Štampači sa lepezom jedini imaju posebari kvalitet otiska za profesionalnu primenu.

Hewlett-Packard 2225

HP 2225 predstavlja prvi komercijalno najavljeni „inkjet" štampač. Odakle mu ovo ime? Radi se o potpuno novom metodu štampanja teksta koji se zasniva na izbacivanju kapljice mastila na papir. Svako slovo je, naime, predstavljeno matricom od 11×12 tačaka. U glavi štampača nema pokretnih delova, to je jednostavno rešeto od 132 otvora iza kojih se nalaze mikro rezervoari sa mastilom i grejači. Kada štampač treba da ispiše neko slovo, neki od grejača zagrevaju mastilo koje se širi i kapljica izleće. Zatim nastupa praktično trenutno hlađenje. Iz većeg rezervoara u glavu biva povučeno novo mastilo a glava se za to vreme pomara do sledeće pozicije na kojoj treba pisati.

U čemu je prednost ovakvog metoda? Pre svega u glavi nema pokretnih delova, pa je štampač jeftin (500 dolara, što je nevarovatno niska cena za firmu kao što je Hewlett-Packard) i, što je možda još važnije praktično nečujan. Ostverena je osim toga i prenosivost — dimenzije HP 2225 su 8,9×29,2×20,6 cm što znači da lako staje čak i u manju tašnu. Napajanje je na struju ili baterije a firma tvrdi da jedan set baterija omogućava štampanje oko 200 (!) strana što znači da je potrošnja više nego mala.

HP 2225 može da se pohvali i mnogim drugim dobrim karakteristikama: karakteri mogu da se definišu na matrici 11×12, u jedan red staju 40, 71, 80 ili 142 slova, što znači da je grafička rezolucija 192×96 tačaka po inču (=2,5 cm) ili 1296 tačaka u redu (može dakle da se štampa slika sa etendardnog televizora ako sa raspolaže pogodnom opremom). Set karaktera obuhvata preko 150 znakova zajedno sa slovima iz danske, škotske i engleske, finske, francuske, nemačke, italijanske, norveške, španske i švedske abecede.

Kada su svojevremeno najavljeni u časopisima mnogi su smatrali da „inkjet" štampači nemaju šanse jer im je potreban skup „repromaterijal". Hewlett-Packard je pokazao da nije baš tako. HP2225 piše na specijalnom ili običnom papiru (u oba slučaja kvalitet je sasvim zadovoljavajući, a trik što se na specijelizovanom eliminiše razlivanje mastila pa je kvalitet uporediv na samo sa matričnim nego i se štampačima sa lepezom) dok se umesto trake menja čitava glava kada se mastilo iz nje istroši. To se događa posle ispisanih 500 strana (rukopis solidne knjige) a jedna glava košta svega 8 dolera — ne skuplje nego traka za matrični štampač!

Čini nam se da HP2225 predstavlja izvanredan izbor, naročito za one koji su odlučili da kupa prenosiv i moćan računar kao što su HP75 ili novi HP71B (HP2225 može da se nabavi sa Centronics, HP IL ili HP IB interfejsom koji je uračunat u cenu od 500 dolara). Kada u bliskoj budućnosti neka druga firma proizvede sličan štampač njihove cene će opasti pa „inkjet" modelima predvidamo veliku budućnost!

Dejan Ristanović

Knjiga koja vas popularno uvodi u svijet kompjutora!

Peter Laurie

KOMPJUTOR U KUĆI

priručna enciklopedija

Najtraženija engleska knjiga o kompjutorima — sada i na našem jeziku!

Iz sadržaja: — OPĆENITO O KOMPJUTORU — PROGRAMIRANJE — PRIMJENA U PRAKSI: Poslovni programi, kompjutor i slikarstvo, simulacije, manipulacija slike, kompjutorska animacija, kompjutor koji govori, kompjutorska glazba, robotika, roboti u industriji, programiranje prilagođeno za kućne kompjutore SINCLAIR itd. — POGLED U BUDUĆNOST — RJEČNIK POJMOVA — SHEMATSKE SKICE I UPUTE itd.

Temeljni priručnik za svakog koji namjerava nabaviti kompjutor za kuću, školu ili za poslovne potrebe

Bitne osobine: jasnoća i preglednost teksta, stručna pouzdanost objašnjenja i bogatstvo korisnog ilustrativnog materijala

Pretplatna cijena: 2.200.- dinara
Prodajna cijena: oko 3.000.- dinara
Izlazi iz tiska: 24. oktobra 1984.

Narudžbenica Računari II

Izdavačkoj jedinici „Cankarjeve založbe"
Ilica 26, 41000 Zagreb

Ovime naručujem _____ primjeraka knjige KOMPJUTOR U KUĆI (izdanje na srpskohrvatskom) po pretplatnoj cijeni od 2.200 dinara što ću platiti u pet mjesečnih rata po 440 dinara.
Knjigu ću primiti po izlasku iz tiska i uplatiti cjelokupne pretplatne cijene.

Ime i prezime naručitelja _____

Adresa _____

_____
Vlastoručni potpis

Napomena: pretplatiti se možete i na telefonski broj (041) — 432 325

PARTNER

PARTNER - SARADNIK U SAVREMENOM POSLOVANJU

ISKRA DELTA uvodi na jugoslovensko i atrano tržište mikroračunar PARTNER, rezultat sopstvenog razvoja. Ne nudimo vam samo mašinsku opremu, nego i savremeno integrisano rešenje vašeg finansijakog, skladišnog i materijalnog poalovanja. Međutim, i pored svega toga, primena je tako jednoatavna da možete da je savladate u jednom danu.



galaksija u školi

Računari i obrazovanje

... u rukama Mandušića Vuka
svaka puška biće ubojita!
(Njegoš: Gorski vijanac)

Brojni podaci

Obrazovanje u oblasti programiranja može biti zasnovano na izradi programa koji rade sa različitim objektima: brojevima, tekstom ili grafičkim objektima. Međutim, ipak se najčešće, za ovu svrhu, koriste brojevi. U bejziku za „Galaksiju" brojevi se u internom kodu registruju u pokretnom zarezu. Međutim za cele brojeve iz intervala −999999 999999 korisnik ima utisak da radi se celim brojevima, jer ih unosi kao cele, e računari ih izdaje u istom obliku. Za ostale brojeve koristi se pozicioni ili eksponencijelni zapis brojeva.

Na ovaj način „Galaksija" omogućuje izučavanje svih elementarnih struktura nad brojnim podecima, koje se javljaju u programu linijske, razgranate i ciklične kao i korišćenje potprogrema. Posebno treba istaći da razgranate strukture koje često u bejziku dovodi do loše čitljivosti programa ovde u naredbi IF ima opciju ELSE što deje bolje strukturirana programa. Teko program

```
10 INPUT X
20 IF X < 0 PRINT X * X  ELSE PRINT X
30 STOP
```

izdaje kvadrat unetog broje eko je uneti broj manji od nule odnosno izdaje uneti broj eko ovaj nije manji od nule.

Od strukturiranih brojnih podetaka postoji samo jednodimenzionalni niz čiji broj elemenate zavisi od raspoložlvog memorijskog prostore. Dakle, posebno sa na dimanzioniše. Za obuku u progremiranju ovo se na može uzeti kao naki bitan nedostatek jednostavno iz razloga što je jednodimanzionalna strukture podetaka zapravo prirodno najbliža fizičkoj strukturi memorije. Višedimanzionalne strukture, kao što su tebele i metrice, mogu se obrazovati nad jednodimenzionalnim nizom i predstavljaju lepe zadatke za vežba iz programiranja.

Od elemantarnih funkcije postoji samo celobrojna funkcije (INT) dok se za trigonometrijeke, eksponencijelnu i logaritamsku funkciju mogu napisati odgovarejući potprogiemi na bejziku. Pisanje ovih potprograma na raznim obrazovnim nivoima može imati različit emisao od programiranje potprograma i izgradnje biblioteke potprograma do numeričkih problema eproksimacije funkcije elgeberskim izrazima u kojima se javljaju samo ertimaličke operacije sebirenja, oduzimanja množenja i deljenja.

Primana računara na rešavanje numeričkih problema može se u školema leko demonstrirati od najnižeg do najvišeg obrazovnog nivoa. Ovde se kao poseban izazov javlja traženje efikasnijih numeričkih algoritama, što „galaksija" omogućuje merenjem vremena rada programa, korišćenjem programski kontrolisanog časovnika. Evo jednostavnog primera: sastavite program koji određuje broj šestocifrenih prirodnih brojeva koji zadovoljeveju uslov da su im zbirovi tri cifre veće i tri cifre manje težina jednaki. Ako progiem zasnujete na ideji da se ispituju svi šestocifreni prirodni brojevi trajeće na računaru oko 15 časova. Međutim, ostevljamo čitaocu da nada etikasniji algoritam koji će sa na računaru izvršiti za samo 75 sekundi!

Ukratko, bejzik na računaru „galaksija" pruža mogućnosti obuke u progiemiranju, kao i u pripremi računara u oblasti jednostavnijih naučno-tehničkih izičeunavenje.

Rad sa tekstom

Praktično pri svakoj upotrebi računara korisnik ima potrebu da u manjoj ili većoj meri radi sa tekstom. Narevno, postoje takve primene računara u kojima se isključivo redi sa tekstom kao što je izučavanje prirodnog jezika, prevođenje sa jednog jezike na drugi u velikom broju informacionih sistema i sl. Rad se tekstom tiebe de omogući • prirodni dijalog korisnika se računalom • uređenje tekste, • pretraživenje teksta i • unošenje i izdavanje teksta.

U interaktivnom radu se računarom korisnik najčešće odgovare na poruke koje izdeje računar. U ovom slučaju tekst služi da podiži komunikaciju računare se korisnikom. Na primer u jednom programu ze obračun kamate računai izdaje pitenja

```
KAMATA U PROCENTIMA? 30
```

e korisnik unosi vrednost kamate u ovom primeru 30% izdavanjem pitenja korisniku se olakšava korišćenje progiema jer bi bez pitenja mogao da pogreši u redosledu unošenja podetaka, što bi dovalo do pogrešnih rezultate u progiemu. Narevno isto tako je važno e možda još značajnije, da računai uz brojeve koji su rezulteti rada progiema izde i tekst koji objašnjava razulteta. Na primer, već pomenuti program za obračun kamate može izdati rezultete u obliku

```
PRETHODNO STANJE  20000
KAMATA            6000
NOVO STANJE       26000
```

Ovo je očigledno znatno čitljivije nego da su izdata samo tri broja 20000 6000 i 26000. Galeksije" u potpunosti omogućuje unošenje i izdavanje teksta u smislu podiške dijeloga između korisnika i računara.

Naki mikroračunarski sistemi su orijentisani na red se tekstom, pa omogućuju uređenje i pretraživanje veće količine teksta. „Galaksija" nije takav sistem iz dva razloga: • u jednom redu na ekranu se izdaju 32 znaka, i • ne postoje mala slova.

Izborom rešenja sa 32 znaka u redu postignuta je dobra čitljivost brojeva i teksta na ekranu bilo kojeg televizora. Ovo je onamogućilo ozbiljniji rad sa tekstom, pa nije imalo smisle uvoditi male slova, jer bi to samo napotrebno poskupelo sistem. Međutim, „galeksija" ipak pruža izvesna mogućnosti koje su značajne u obuci progiemiranja a odnose se na rad sa tekstom. Teko se može višiti dopisivanje tekste (konkalenacija) i ispitivanje na jednakost. Neka delje mogućnosti mogu biti ostvarena izgiednjom biblioteke potprograma. Tako sledeći potprogiam vrši poredenje dva ezbučna (znakovna) podatka koji eu eusedni elementi azbučnog niza.

```
3000 FOR K=0 TO 15
3010 A=BYTE (PTR X$(I)+K)
3020 B=BYTE (PTR X$(I+1)+K)
3030 IF A+B ELSE K=0 RETURN
3040 IF A < B K=  1 RETURN
3050 IF A < B K=1 RETURN
3060 NEXT K K=0 RETURN
```

Rezultat poređenja se postavlja kao vrednost promenijiva K na sledeći način K dobije vrednosi −1. O ili 1. prema toma da li je X$(I) iedom manje jednako ili veće od X$(I+1). Gornji potprogiem se može uspešno koristiti za uređenje azbučnih nizova podataka u azbučni redosled. Što svakako treba de je prisutno u svakom kursu iz progremiianje.

Grafika na računaru

Većina mikroračunarskih sistema ima određene grafičke mogućnosti koje se sastoje u crtanju na etandardnom ekranu crno-belog ili kolor televizore. Crtenje se postiže tako što se izvestan broj tačaka na ekienu nalazi pod progiemskom kontiolom. Prema toma koliki je ovaj broj tečaka gratika se deli na grubu i finu grafiku. U grubo grafici progremaki se kontioliše nekoliko hiljada tačaka • u finoj nekoliko desetina hiljade tečaka. pe i više. Galekalije raspolaže grubom gretikom se 64×48=3 072 tečke na ekrenu. Iako su ovo vilo skromna gretičke mogućnosti njihovo postojenje čini „galeksiju" znatno interesantnijom i u obrazovanju iz sledečih razloga • daje mogućnost upoznavenja učenika se gretičkim mogućnostima računare • znatno proširuje tond zadeteka interesantnih za progremirenje • daje mogućnost progiemirenja nekih video igara što posebno doprinosi interesovanju mladih za programiienje.

Da bismo ilustiovali kako i vilo jednostavna rešenja mogu da proizvedu intere

Mikroračunari nalaze sve veću primenu u raznim oblastima. Jedna od vrlo značajnih primena mikroračunara jeste i obrazovanje. Na svetskom tržištu postoji bogat izbor „obrazovnih" paketa, ali je cena ovih uređaja veoma značajna, posebno u našoj sadašnjoj ekonomskoj situaciji. Može li, međutim, biti kakve koristi u školama i od mikroračunara minimalnih mogućnosti? Ovoj klasi pripada i mikroračunar „galaksija", čiji skromnih hardverskih potencijala, „galaksija" omogućuje rad sa brojevima, tekstom i grafikom. Koje su njene stvarne mogućnosti kao učila za programiranje, a koje kao sredstvo za unapređenje nastave?



Prva lasta. Sa računarom „galaksije" počinje da se ostvaruje ideje o uvođenju računara i u naše škole

santna crteže na ekranu, posmatrajmo sledeći program

```
100 iDEMONSTRACIJA GRAFiKE
110 HOME
120 X=INT(32*RND)
130 Y=INT(24*RND)
140 FOR I=X TO 63-X
150 FOR J=Y TO 47 Y
160 IF DOT I ,J UNDOT I,J ELSE DOT I,J
170 NEXT J NEXT I
180 FOR I=1 TO 2000
190 IF KEY(4) GOTO 220
200 NEXT I
210 GOTO 120
220 FOR X=0 TO 63
230 FOR Y=0 TO 47
240 IF DOT X, Y UNDOT X Y ELSE DOT
X Y
250 NEXT Y NEXT X
260 GOTO 260
```

Program bira slučajnim izborom tačku u levoj gornjoj četvrtini ekrana, a zatim koristeći ovu tačku kao jedno tema pravougaonika, crta centralni pravougaonik na ekranu. Prevougaonik ispunjava tako da „pali" ugašene tačke a „gasi" upaljene tačke. Posle iscrtanog svakog prevougaonika program čeka 18 sekundi a onda bira novu slučajnu tačku itd. Ako korisnik želi da zadrži sliku u peuzi između crtanja pravougaonika treba da pritisna taster D posle čega nastaje završna obrada slike i program se završava. Pri svakom izvršavanju programa dobija se drukčija slika na ekranu.

Mašinski jezik

Profesionalno obrazovanje u oblasti računarstva ne može se u potpunosti ostvariti bez prisustva mašinskog nivoa programiranja. Zaprevo tek upoznavanjem mašinskog jezika i arhitekture konkretnih računara na previ način se sagledavaju i mogućnosti računare. U slučaju Galeksije' može se upoznati mašinski jezik mikroprocesora Z80A koji je inače često korišćen osmo bitni mikroprocesor. Manji programi na mašinskom jeziku mogu biti pisani u heksadekadnom obliku. Unošenje ovakvih programa najbolje je organizovati u obliku programske datoteke. Na ovaj način će se leko vršiti ispravke u programu. Ilustrujemo ovo sledećim primerom

```
10 FOR I=&2C3A TO &2C48
20 TAKE A
30 BYTE I,A
40 NEXT I
50 INPUT X$
60 BYTE &2C41 BYTE(PTR X$)
70 X=USR (&2C3A)
80 GOTO 60
90=&01 &OO, &02 &21 &00 &28 &36
&41
95=&23 &OB &CB &76, &28, &F8, &C9
```

Pre unošenje programa u memoriju treba primeniti komendu NEW 15 da bi se oslobodio prostor za smeštaj potprogreme na mašinskom jeziku. Startovanjem programa vrši se prenošenje mešinskog potprograma iz programske datoteka u memorijski prostor od adrese &2C3A do &2C48, a zatim uneti znak sa tastatura (neredba 50) postavi se u edresu &2C41 i prede ne izvršavenje mašinskog potprograme koji upiše uneti znak tako da se izda u svim pozicijeme na ekranu.

Pisanje ozbiljnijih programe na nivou mašinskog jezika zahteva simbolički jezik, e za ovo je neophodno raspoiegati odgovarajucim asemblerom za „galeksiju". Razvoj ovakvog asimbiere nije veći programski probiem pa će, nasumnjivo, uskoro biti na respoieganju i za „galaksiju".

Proširenja „Galaksije"

Većine mikroračunarskih sistema je otvorena prema raznim vrstama sotiverskog i hardverskog proširenja pa i računar „geleksije". Osnovna varijanta „galaksije" može se programskom nadgradnjom učiniti močnijom za razne primene. Ovo se leko postiže izgradnjom biblioteka potprograma koje podržavaju neke specifičnosti pojedinih primena, kao što je bibliotaka lementarnih matematičkih funkcije, red se tekstom gratički red, kao i izrede najraznovrsnijih aplikacionih programa, posebno za potrebe obrazovanja.

Hardverska proširenja treba orijentisati pre svega na ona delove koji baziraju na elektronskim komponentema kao što je proširenje RAM-memorije, fina grafika, ton i sl. Veza računare sa spoljnim svatom, koje omogućuje primenu u menjim eksperimentalnim situacijema, bila bi tekoda vrlo značajna u našim obrazovnim sadržajima. Posebno bi, svakako bilo korisno povezivanje električna pisače mašina sa „galekaijom" pre svega za potrebe štampanja programa.

Prvi korak

Osnovne varijanta „galeksije" sa 4 KB ROM i 4 KB RAM-memorije predstavlje mikroračunar koji u obrazovanju obezbeduje

- zbližavanje učenika sa računarom
- obuku u progremirenju na bejziku
- upoznevanje grafičkih mogućnosti računara na nivou grube grafike
- upoznavanje mašinskog nivoa progremiranja
- obuku u programskim tehnikama koje su od značaje za navedena nivoe programirenja.

Prema tome, svi kursevi informativnog karaktere iz oblasti računarstve bez obzire ne obrazovni nivo, mogu se zasnivati na „geleksiji". Kade učenik sevleda programiranje na „galeksiji" može pisati programe za prenošenje stečenih znanja iz raznih predmata ne računar. Ako bi profesori i bili u stanju da postava ovakve programske zadatke učenicima i de nadgleđaju njihov red, to bi bio izvanreden način za kreativnu i kritičku primenu stečenih znanja.

Autor ovog teksta varuje da bi ovakva primena računare u školema doprinala motivisanosti učenika za sticenje novih znanja e tima i unapređenju nastava. Iz opštih i stručno-obrazovnih predmata. Pisanje programa je kreativan posao u kojem projektant programa unosi sva svoja znanje o odredanoj temi i razultate svog rade daje na korišćenje drugima. To je intelektualna aktivnost čije je obrazovna vrednost najveća za ona koji pišu programe, a znatno manja često i nikakva za ona koji samo koriste rezultate rede tudih programa. Galeksije pruža mogućnost da se prvi korak u primeni računare u nastavi učini, najzad i u našim školema.

Prof. dr Nedeljko Parezanović

moj prvi program

Programiranje za početnike

Škola programiranja u 100 lekcija

Šta najpre znači „znati bejzik"? Pod ovim terminom se podrazumeva poznavanje određenog broja naredbi i njihovog dejstva, razumevanje strukture programa i potprograma i uopšte, sposobnost de se deli algoritam pretvori u program za neki računar. No, programiranje je nauka koje sa nikako ne može svesti ne pretvaranje gotovog algoritma u program — veština je napraviti algoritam ili kada daije nepredvidjivo pronaći problem koji može algoritamski da sa reši pa ga onda rešiti. Ova Škole programiranje ne može obzirom na prostor koji zauzima, da zameni čitave biblioteke knjiga posvećene onime koji uča programiranje profesionalno ili amaterski. Njen zadatak je kroz sastavljanje jednog programa koji će prolaziti kroz nekoliko etapa de vas podstakna na dalji samostalni rad i istraživanje. Ne treba, kao i obično, da očekujete de ćete program koji sastavimo moći koristno da upotrebite — koristen program može da bude toliko složen de nije pogodan za početnike. No ukoliko budete pažljivo čitali sledeće redove verovatno ćete sa prisetiti nekog problema iz vašeg svakodnevnog života koji može da se reši primenom računare i skupiti hrabrosti da se posvetite njegovom rešavanju. Ukoliko sa to zeiste dogodi, ova Škole je potpuno ispunile svoj cilj.

Programi koje budemo pisali biće namenjeni računaru gelakaije pa će njeni vlasnici imati lepu privilegiju da Ih ukucaju bez ikakvih izmena. No Škole je u suštini nezavisna od računara pa ćete moći de je pratite i ako ste vlasnik Spectrume ZX81 pa čak i (premde uz malo problema) nekog džepnog računara sa mogućnošću programiranja. Ukoliko imate probleme sa razumavenjem neke naredbe koje vam izgleda kao spacijalitet računare galakcije uzmite uputstvo za njegovu upotrebu (jedan od rezloga što smo se opredelili za ovej računar je u tome što svi posedujete njegovu dokumanteciju) i utešite se time da se prevodenjem programa mnogo što-šta nauči.

Ukoliko smatrate da ste dobro upućeni u programiranje, možde ćete ipak poželeti da pročitate ovu Školu. Algoritmi i tehnika koje smo koristili su za početnike dobri jer su jednostavni. U mnogim slučajevima postoje bolje rešenje — ne vema je da ih pronadata

Čas 001 — Algoritmi i programi

Pre nego što počnemo sa pisanjem prvog programa moraćemo da pogledamo kako sve to izgleda na daleko jednostavnijim primerima. Teko će, doduše, naš prvi program biti neš treći program ali — nigda ne žurimo.

Reč „algoritam" potiče od imena uzbeskog (autor ovog tekata mora da prizne da mu reč „uzbeskog" ne zneči beš mnogo) matematičara Al Horezmije koji je živeo u IX veku naše ere. Ovaj matematičar neobičnog imena je definisao pravile za izvršavanje četiri osnovne računske radnje (pravilo za deljenje je bilo znetno primitivnije od onoga koja denas poznaju učenici nižih razreda osnovna škole!) i teko postao začetnik jedne danas neobično popularne grane matematike koja se naziva „teorije algoritema".

Verovatno je za većinu naših čitalaca potpuno besmislena egzaktna definicije algoritma koju matematičari danas usvajaju (samo potpunosti radi najmanje strašna definicije algoritme je „skup instrukcije koje može da izvrši Postova mašine"). Ipak pojam algoritma je veome lako shvatljiv svakoga dana izvršavamo hiljade algoritama a da toga uopšte nismo ni svesni. Jeste li na primer nakade čekali autobus 84? Ako jeste, evo algoritme koji ste tade primenjivali:

1 Dodi na stanicu
2 Da li je na stanici neki autobus? Ako jeste predi na korak 3 a eko nije ponovi korek 2
3 Pogledej broj autobusa
4 Ako je taj broj 84 predi na korak 5. U protivnom vrati se na korek 2
5 Udi u autobus
6 Kupi kertu
7 Poništi kartu
8 Da li se autobus kreće? Ukoliko se kreće vrati se na korak 8 a ukoliko miruje — predi na korek 9
9 Da li autobus stoji na stanici? Ukoliko je odgovor da predi ne korak 10 u protivnom vrati se na korak 8
10 Da li je to stanica na kojoj treba de izađeš? Ako nije vrati se na korak 8 a ako jeste — izađi iz autobusa
11 Stigao si na cilj

Kao što vidimo algoritmi se formulišu po tačkama (koracima). Svakoj od njih je obavezno dat broj na koji se drugi koraci ako je to potrebno pozivaju. U okviru sveke tačke dato je neko precizno uputstvo koje trebe izvršiti. Za formulisanje algoritma neobično je važno de poznajemo naredbe koje onaj kome je uputstvo namenjeno (računar, robot, drug kome objašnjavamo gde stanujemo ...) ume da izvrši. U gornjem primeru korak 7 glasi „poništi kartu". No moguće je da onaj kome je algoritam namanjen jednostavno ne zne šta to znači. Njemu bi bilo nužno objasniti keko da pronade automat za poništavanje karata, kako de okrene kartu pre nego što će je staviti u prorez automate, koliko treba de čeka dok se automat ne oglasi zvonom šta de radi ako automat na funkcioniše. Obzirom da nas najviše zanime pisanje programa za računare, algoritemski koraci će najčešće biti ekvivalentni naredbama nekog programskog jezika u našem slučaju bejzike.

Vreme je da napišemo i mali program. Obzirom da ne posedujemo robota koji bi putovao autobusom 84, moraćemo da se okrenemo nečemu mnogo prozaičnijem — brojevima. Možde ste čuli za problem nezvan Ulamovo nagadanja (Ulem's Conjecture!) Ako niste, pomoći ćemo vam da nagadate i to uz pomoć računara.

Evo najpre algoritma koji je za početak napisan samo rečima. Ulazna veličina je bilo koji prirodan broj n.

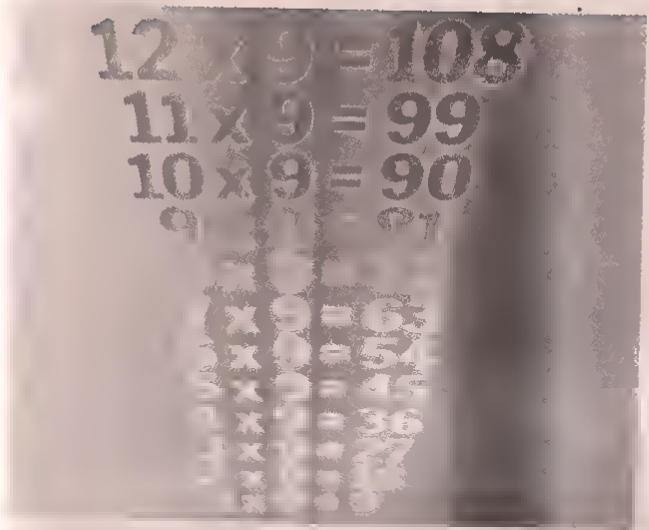
1 Da li je broj n paran? Ako jeste predi na korak 3 a ukoliko nije na korak 2
2 Pomnoži n sa tri i dodaj jedan razultatu. Zbir nazovi n (time ja stara vrednost n zeboravljena!) Predi ne korak 1
3 Podeli n sa dva i količnik ponovo nazovi n
4 Da li je n jednako jedan? Ukoliko nije predi na korak 1 a ukoliko jeste završi sa radom

Algoritam je prilično jasan pe reč na gadanje izgleda nije pogodna za njegovo ime. No nije tako. Vidi se da se polazi od broja n i da se on neprekidno deli sa 2 ili množi sa tri, pa se rezultatu dodaje jedan) sve dok se ne dobije broj 1. Pitanje je da li će bez obzira na broj n od koga smo počeli uvek na kraju da se dobije 1 ili postoji neki broj n za koji jedinica nikada neće biti dostignuta. Poljski profesor matematike Ulem je pretpostavio da se za svako polazno n na kraju dobija 1 ali mu nije pošlo za rukom da to i dokaže što ber za sada nije uspelo ni jednom drugom matematičaru (za egzaktno rešenje problema je i dalje raspisena privlačna novčana nagrada) pa je ostalo samo da se — negeds. Sastavimo dakle program koji transformiše neko n u broj 1 i pokušajmo da pronadamo neko n za koje se to neća dogoditi!

Na početku rada trebe da zatražimo od korisnika da unese neko n. Za to nam jasno služi naredba INPUT. Zatim trebe da ispitamo da li je n deljivo sa dva (broj je paran ako je deljiv sa dva). To ćemo učiniti lako što ćemo n najpre podeliti sa dva i neći ceo deo rezultata koji ćemo nazveti m. Zatim ćemo ispitati de il je m jednako n/2. Ukoliko je odgovor na ovo pitanje potvrdan, n je paren broj (ne primer INT (4/2)=4/2=2) a eko je odrečan — n je neparno (npr. 2=INT(5/2)≠5/2=2.5). Ostatak algoritma je prilično jesan — posedujamo bejzik naredbe koje odgoveraju pojedinim algoritamskim koracima.

```
10 INPUT N
20 M=INT (N/2
30 IF M=N/2 GOTO 60
40 N=3*N+1
50 GOTO 20
```

„Osnovna škola bejzika u stripu", koju smo objavili u našem prvom specijalnom izdanju o računarima, bila je namenjena onima koji u programiranju čine prve korake. Iako takvih sigurno ima dosta, još je više onih koji su upućeni u osnove bejzika ali ne znaju šta bi dalje radili sa njim. Ovo poglavlje je namenjeno njima.




```
60 N=N/2
70 IF N=1 PRINT . DOSTIGNUT BROJ 1
ELSE GOTO 20
```

Čim startujamo program uobičajenim RUN računar će ispisati znak pitanja tražeći od nas da unesemo neki broj. Učinimo to i pritisnimo ENTER (ili RET — zavisi od tastature koju imamo) i nekoliko trenutaka posle toga primetimo da je računar završio posao i na to nas upozorio odgovarajućim komentarom. Ovim je ujedno završen i čas 001. Na kraju svakoga časa u pravoj školi su dopuštena pitanja. Mi jasno, ne možemo da predvidimo sva pitanja koja će čitaocima ove „Škole" pasti na pamet, ali možemo da odgovorimo na ona koja sa nameću sama od sebe.

— Ja ne verujem računarima. Sve što ovaj program radi je da posle dužeg ili kraćeg vremena prikaže neki komentar na ekranu. Da li bi bilo moguće modifikovati program tako da računar prikazuje brojeve u nizu sve dok na prikaže broj 1?

• Nema većih problema, dovoljno je dodati naredbu 25 PRINT N i računar će u svakom prolasku kroz ciklus prikazivati broj iz niza.

— Da probamo. Radi ali brojevi prebrzo prolaze — i dalje ne mogu da provarim da li računar dobro radi.

• I to može da se reši i to na dva načina, ukoliko, u toku izvršavanja programa pritisnemo i držimo DEL, sve će se zamrznuti. Ako ne želimo da se mučimo na ovaj način dodaćemo naredbu 26 FOR I=0 TO 1000: NEXT I i računar će čekati posle prikazivanja svakog broja. Od interesa bi međutim, bilo i prebrojati koliko ima prolazaka kroz ciklus, bilo potrebno da se dođe do kraja posla. Kako bi to moglo da se ostvari?

— Trebalo bi dodati jednu promenljivu koja bi se povećavala za jedan pri svakom prolasku kroz ciklus. Neka to bude promenljiva P, dodajmo dakle, naredbu 27 P=P+1 i na kraju programa 35 PRINT P. Neka N bude 5. Dobili smo da ima računar prošao kroz ciklus 4 5 puta! Da li će možda poslednji prolazak računat polovično?

• Naravno da ne. Jednostavno, promenljiva P na početku rada nije imala vrednost 0 već 0 5 (kod drugih računara bi možda bila prijavljena greška: No such variable at line 27). Grešku tako ispravljamo dodajući naredbu 5 P=0.

— Da li je neko već eksperimentisao sa većim brojevima? Da li se uvek dobija rezultat 1?

• Pomoću velikih kompjutera isprobani su brojevi sa deset petnaest i u poslednje vreme dvadeset cifara. Pokazalo se da svi oni posle duža ili kraća konvergencija dolaze do broja 1. To naravno ne znači da jednoga dana neće biti pronađen broj koji se ne transformiše u 1 na opisani način, ali daje realna osnove za pretpostavku da je nagađanja prof. Ulama tačno.

— Da li naš program radi za tako velike brojeve (do 1 E38)?

• Veliki broj je možda neparan. U tom slučaju računar ga množi sa 3 i rezultatu dodaje 1. Ukoliko je zbir opet neparan operacija se ponavlja, što znači da je program već udesetostručen. Može dakle lako da se desi da broj izađe iz opsega u kome računar radi i da se pojavi greška pri zaokrugljivanju. Zato broj koji kucamo treba da bude bar 20 puta manji od maksimalnog celog broja koji galaksija može da prihvati (vama ostavljamo da pronađete koji je to broj).

— Znači li to da računar galaksija ne može da nam pomogne ako želimo da ispitujemo veće brojeve?

• Naravno da ne znači. Mogli bismo da smeštamo svaku cifru broja u elament numeričke matrice A(t) i da sami sintetišemo računske operacija. No to je problem kojim ćemo sa zabaviti nekom drugom prilikom. Shvatili smo šta su program i algoritam, pa smo spremni da se uhvatimo u koštac sa problemom koji ćemo rešavati u ovoj Školi.

Čas 010 — Tablica množenja

Prethodni problem je bio jednostavan zato što nam je algoritam bio unapred dat i treba ga samo pretvoriti u program. Došlo je vreme da se oprobamo i u sastavljanju algoritma. Treba da napišemo program pomoću koga računar testira korisnikovo poznavanje tablice množenja. Ovaj program naravno nije namenjen nama koji ćemo ga pisati — znanje bejzika podrazumeva i neka najosnovnija matematička znanja. No za osnovce iz naše okoline ovaj program može da bude zabavan i što je još važnije, korisian.

Pre svega, treba jasno da definišemo problem i odlučimo šta računar treba da radi. Testiranje tablice množenja se obavlja tako što računar zadaje probleme (probleme su tipa 2*7=?) a zatim očekuja od korisnika da ih reši. Korisnik kuca odgovor, računar provarava njegovu ispravnost i izdaja odgovarajući izveštaj. Proveravanje rezultata ne bi trebalo da predstavlja poseban problem — svaki računar zna da množi a naredba IF je kao stvorena za testiranje rezultata. Pravi problem je dakle, postavljanje problema. Kako računar može da odluči koje će brojeve prikazati korisniku?

Prvo što ćemo pomisliti je da se u memoriju računara stave svi mogući parovi brojeva. Obzirom da će naš program testi

rati poznavanje tablice množenje 10x10. pamćenje 100 parove brojeva bi odnalo 800 bajta memorije što i nije previše. U memoriju bismo upisali i redosled postavljanje problema prvo na primer računar treba da prikaže osmi par brojeva zatim pedseat šesti i tako dalje. To bi bduzeto još izvesten prostor eli bi utrošak memorija i datja bio prihvatljiv No računar bi zadevao probleme uvek istim redom što znači da bi test posle Izvesnog vremena, posteo nepouzden U takvom tranutku bismo morali da menjamo čitev program ugrađujući u njega novi redosled brojeva To, po mogućstvu, treba de izbegavamo — jednom napisani program bi morao de bude u stanju da pouzdano radi bez Ikakvih modifikacije

Ideje koju smo Imali dakle nije dobra. Kako da dodemo do dobre ideje? Najbolje je de razmislimo o načinu na koji bi čovek rešavao lati problem. Kako, na primer, učitelj testira poznavanje tablice množenja? Tako što kaže neka dva broja i bd učenika zahteva de ih pomnože. Kako on dolazi do ta dva broja? U većini slučajeva kaže prve brojeva koji mu „padnu na pemet" — ti brojevi su dakle slučajni.

Pojem slučajan broj se vrlo teško može detinisati. Što se nizova slučajnih brojeva tiće njihova definicija je prilično jednostavna: to je grupa brojeve koji nisu povezani nikakvim medusobnim relecijama. Ukolko ste proučili uputstvo za upotrebu vešeg računare i našu Osnovnu školu bejzika svakao ela prmetili da je i računar galeksije snabdeven naredbom RND koje služi za generisanje stućejnih brojeva. Kako je uopšte moguće da računer generiše slučajne brojeve kade su svi rezulteti koje on deje stroga funkcije ulaznih vellćina?

Slučajni brojevi koje računer generiše su u stveri pseudoslučajni. Medu pseudoslučajnim brojevima postoje nake tajna veze eli su ona tolliko složene da korisnik ne osnovu prethodnih na može nikako da pogodi dalje članova niza — za njega su dakle pseudoslučajni bojevi dovoljno slučajni. Formule koje povezuje članove niza pseudoslučajnih brojeva je najčešće tako konstruisana da se stelno dobijaju brojevi izmedu 0 i 1 i to teko da je njihove srednja vrednost praktično 0.5 (što više brojeva generišete to će njihova srednje vrednost biti bliža ovoj cifri). Za formiranje ovoga niza potreban nam je njegov prvi član de bi iz njaga bio formiran drugi, iz njega treći i teko dalje. Tej prvi član niza se u strenoj literaturi alikovito naziva seed (seed=seme) i generiše teko što računer u nekom tranutku analizire sedržej svog internog časovnika ili nekog drugog registre koji se često menje. Računar galeksije nije nikakav izuzetak — sveki put kade ga uključite PRINT RND će dati drugi rezultat.

Slučajne brojeva u programu dakle generišemo naredbom RND. Možemo da napišemo A=RND i promenljive A će dobiti neku slučajnu vrednost izmedu 0 i 1. Naša tabtica množenje treba medutim de tsetire množenje celih jednocifrenih brojeva. Zato ćemo dobijeni slučajni broj pomnožiti sa devet uzeti ceo deo rezultata (tako dobijemo ceo slučajen broj izmedu 0 i 8) i dodeti mu 1 (broj je sada izmedu 1 i 9) na isti način ćemo da generišamo i drugi slučajen broj a zetim ćemo pomoću naredbe

```
 10 !    TABLICA MNOZENJA
 20 !
 30 T=0
 40 N=0
 50 P=0
 60 HOME
 70 PRINT AT 256,'TACNI';AT 265,"NETACNI",AT 276,"BEZ ODGOVORA"
 90 PRINT AT320,T;AT 331,N;AT 345,P
100 PRINT AT 0,"KOLIKO JE'
110 A=INT(RND*99+1) B=INT(RND*99+1):C=A*B PRINT
120 PRINT AT 9,A," *",B,"="
130 INPUT X$
140 IF VAL(PTR )X$)=C GOTO 300
150 N=N+1 PRINT AT 0,"NIJE TACNO!"
240 PRINT AT 32,A," *",B," =",C
250 FOP I=1 TO 500 NEXT I
260 GOTO 90
300 PRINT AT 0,"B R A V O !"
310 T=T+1
320 GOTO 250
```

PRINT, ispisati problam ne ekranu. Evo i programa:

```
10 A=INT (RND*9)+1
20 B=INT (RND*9)+1
30 PRINT ' KOLIKO JE";A;"*";B;"?"
40 INPUT C
50 IF C=A*B GOTO 90
60 PRINT " ODGOVOR NIJE TAČAN"
70 PRINT " POKUŠAJ PONOVO '
80 GOTO 30
90 PRINT " BRAVO! TAČAN ODGOVOR"
100 GOTO 10
```

Ovaj program se neprakidno izvršava računar zadeje probleme i daje informacije o tačnosti odgovora. Kako sve e neročito uvežbavenje tablice množenja mora na kreju de dosadi treba znati kako se prakide izvršavanje progreme. Dovoljno je pritisnuti taster BRK (BREAK) i računar će ispisati uobičajano READY. Nešto rediklnije rešenje predstevlja pritisak tastere RESET (posledice su iste) ili isključivenje računere iz mreže za napajenje, čime će i program biti izgubljen i nije neki program, reći ćete, eli ga ipak nemojte brisati — redićemo još na njemu. Pitenja?

— U programu se dva pute ponevtja naredba A (ili B)=INT (RND 9)+1. Da li može de se upotrebi potprogram?

* Ne može da se upotrebi potprogram koji bi racionelnije trošio prostor. Neki računari omogućavaju definisanje funkcija što znači de bi naredbe 10 i 20 mogle da se zamene se

```
10 A=FNRNDN
20B=FNRNDN: trebalo bi jasno dodeti i
1000 DEFFNRNDN=INT (RND*9)+1 uštede
```

da memorije, dekle ne bi bile značajna e programm ne bi bio ništa rezumtjiviji.

— Čini mi se de program ovako kako je napisan nije dovoljno enimirejući za dete i de na daje realne rezultate koji bi govorili o njegovom znanju. Progrem pra svega postavlje više problema eli ne prabrojava tačne i pogrešne odgovora. Osim toga vreme za razmišljenje nije nikako ogrenićeno pa onaj ko sedi pred računarom na mora ni de odgovare na pitanje — ne postoji način de sa-ustenovi kolliko je probleme bilo postevljeno!

Tačno. Medutim, ovej program predstevlja samo početak. Pri pisanju nekog složenog programa možemo de se opredetimo za jeden od dva pristupe: de pišemo onakav program kakav on na kraju i treba de bude ili da ga razvijemo korak po korek. Prvi pristup primenjuju iskusniji progremeri koji cena svoje vreme. Drugi je pristupačen korisnicima koji tek počinju razvijajući program oni imaju priliku da rešavaju probleme na koje nailaze jedan po jedan kako nallaze ne njih i da se, kada na mogu dalje da nepreduju, zeustave na nekoj verziji programa i progtase je za konačnu. Radićemo dakle dalje na progremu za testiranje poznavanje tablice množenja rešavajući probleme iz postavljenog pitanje obrnutim redom.

Čas 011 -~ kako ispisati tabelu?

Poslednje primedba je bila de program ne formire tebelu tačnih i netečnih odogovore i rubriku koje govori o vremenu razmišljanja korisnika. Tu tabelu ćemo brzo napraviti ali moramo de se odlučimo ne koji deo ekrena da je smestimo i kako de je formiramo. Najjednostavnija je da se u prvom redu ekrene nelazi problem koji dete treba da rešave u drugom odgovor koji dete kuca a, nešto ispod loga, labele koje ilustruje njegovo znenje. Neke ona počinje od devetog reda na čiji početak kao što vidimo sa mape ekrena (strane 11 uputstva ze upotrebu „galaksije") dolazimo sa PRINT AT 256.

Na samom početku programe moramo de, kako bi to progremeri rekli, inticijetizujemo broječe. To će jednostavno biti promenjilve T N i P. Promenljiva T će brojati tačne odgovore promenjive N netačne e promenljive P probleme koje dete nije rešilo u predvidenom vremenu. U početku problema nije ni bilo pa će sve tri promenljive dobiti vrednosti nula za šta eu nam potrabne tri naredba .

— Da li su te naredbe zaiste neophodna i onima koji nemeju galaksiju? Zar promenljiva T N i P, obzirom da nisu korišćene već nemaju vrednost nuta?

* Kod „galaksije" ne sme de se računa sa početnim vrednostima promenljivih (iskusniji korisnici će primetili da sve promenljive na samom početku reda imaju vrednost 0.5i da se ona ne menje naredbom RUN). Neki računeri (TRS 80 na primer) autometski dodeljuju vrednost nuie svekoj promenljivom kojoj ntje dodeljena vrednost. Čak i eko posedujete računar kod promentjivoj kojoj nije dodetjena vrednost. Čak i ako posedujete računar kod koge dodeljivanje vrednosti promenljivime na početku rada nije neophodno progremerski „bdn-ton" zehteva dege izvršite jer će lako onaj ko analizire pogrem na samom početku znati koliko promenljivih ima koliko će ona memorije zeuzeti i uopšte šta de

```
30 T=0
40 N=0
50 P=0
60 HOME
70 PRINT AT 256,"TACNI";AT 265,"NETACNI";AT 276,"BEZ ODGOVORA"
80 FOR L=1 TO 20
90 PRINT AT 320,T;AT 331,N;AT 345,P
100 PRINT AT 0,"KOLIKO JE"
110 A=INT(RND*99+1):B=INT(RND*99+1):C=A*B:PRINT
120 PRINT AT 9;A;" *";B;
130 Y$="00:00:00:":DOT*
140 INPUT X$
200 S=VAL(PTRY$+6)
210 IF S<25 GOTO 270
220 PRINT AT 0,"PREDUGO RAZMISLJAS!"
230 P=P+1
240 PRINT AT 32;A;"*";B;"=";C
250 FOR I=1 TO 500:NEXT I:NEXT L
260 GOTO 260
270 IF VAL(PTRX$)=C GOTO 300
280 N=N+1:PRINT AT 0,"NIJE TACNO!"
290 GOTO 240
300 PRINT AT 0,"B R A V O !"
310 T=T+1
320 GOTO 250
```

radi eko njegov računar ima strožiji bejzik

Da nastavimo Svaki put kade dete da ispravan odgovor računar treba da izvrši naredbu T=T+1 i tako poveća sadržaj promenljive T za jeden Kada dete pogreši računar će povećati sadržaj promenljive N a kada prekoračí predviđeno vreme — promenljive P Posle toga računar će izvršiti naredbe kojima će stari sadržaj tablica biti zamenjen novim koji je ekteten

— Da li je potrebno da najpre izbrišemo stari sadržaj tablica?

* Nije Koristićemo naredbu PRINT AT koja štampa ono što želimo na proizvoljnoj poziciji ekrena, na obračunajući pažnju na to što se ovim ispisivanjem menja sadržaj koji se na toj poziciji već nalazio U ovom slučaju ne može da se javi problem koji bi u nekim sličnim programima bio neprijetan obzirom da se broj tačnih pogrešnih i zakasnelih odpovora nikada ne smanjuje. ne može da se desi da od prethodnog sadržaje preostane neko slovo koje bi zajedno se novoispisanim promenilo smisao poruke (ako preko broja 10 štampamo broj 9 dobićemo 90 a ne 9)

Programm se prve slike omogućava osim svega što i prethodni. delimično ispisivanje tabela Broj zakasnelih odgovore je. međutim uvek nula. jer još nismo ni govorili o naredbame kojima bismo obezbedili da galaksija meri utrošeno vreme

— To bar nije problem. Pa „galeksije" poseduje ugrađeni časovnik

* „Galeksija" zeista poseduja veome tačan časovnik koji ćemo u ovoj prilici svekako koristiti Da sa podsetimo časovnik je usko povezan sa promenljivom Y$ kojoj na početku traba dodaliti vrednost 00 00 00 Da aktiviramo časovnik koristićemo nerdebu DOT*, e da ga zaustavimo UNDOT* Ove dve naredbe ne treba nikako mešati sa naredbame za crtenja po ekranu — zvezdica na kraju im je sesvim izmenila smiseo Sve bar za sada, izgleda vrio jednostavno na početku ektiviramo časovnik i dodalimo mu nuitu vrednost a zatim pošto dete de odgovor provarevemo da li je prošio više od na primer 25 sekudni Problem međutim leži baš u ovom poređenju kako de uporadimo broj proteklih sekundi sa 25?

— Ništa lakše upotrabićemo naredbu IF Y$25 P=P+1

a Tamen posle! Pre svega Y$ je elfanumerik a 25 je broj pa nema nikakvog smisla poređiti ih Mogli bismo naravno, da napišemo I IF Y$>,25 eli ni to ne bi bilo ništa bolje Pre svega u Y$ nije smešten samo broj sekundi već i broj časova i minuta u formatu koji smo maločas opisali Čak ni IF Y$>,00 00 25" ne bi mnogo pomogio jer galeksije ne omogućave poređenje alfanumerika uz pomoć simbole > i < — moguće je samo proveravatt de li su sadržaji dve elfanumeričke promenljive jednaki i to pomoću funkcije EQ

— Problem dekle ne može de se reši .

* Neravno da može „Galakaije", rekosmo ne moža de poredi stringove ali sasvim lepo poredi brojeve Potrebno je dakle da pretvorimo sadržaj Y$ u broj sekundi

— Pa de! Za to služi neredba TIME opisane u pogievlju koje je nemenjeno iskusnijim viesnicima „galeksije Samo

* Samo što se radi o programu na mašinskom jeziku koji čak ni onima koji čitaju to poglavlje nije baš sasvim jesen! Ne mi ćemo problem rešiti koristeći „galeksijin bejzik On poseduja naredbu VAL koje pretvare alfanumerik u konstentu Ukoliko kako piše u uputstvu ze upotrebu upotrebimo naredbu A=VAL (PTR Y$) nećemo uraditi ništa naročito. Promenijive A ce zarsta dobiti vrednost konvertovanog stringa u Y$ ali samo do simbola koji ze galaksiju služi kao ozneka kreje izraza Dobićemo dakle broj sati koji će uglavnom uvek biti nula Nas interesuju sekundi koji su sedmi i osmi simboi promenljive Y$ Zato ćemo upotrebiti VAL (PTR Y$+7)

— Zer smemo de sabiremo Y$ (alfanumerik) sa 7 (broj)?

* Nerevno de ne ememo Međutim, ispred Y$ sa nalazi naredba PTR koje kao što znamo daje edresu promenljive u memoriji (u našem slučju, to je 10880 što bismo videli iz mepa sistemskih promenljivih de znemo de pretvorimo broj 2A80 u dakadni zapis) tj broj Tom broju dodajemo sedam što je sasvim regulerno e onda izračunavamo vrednost izraza koji je na tu adresu smešten

— Mogli smo dekle da upotrabimo i naredbu A=VAL (10887)!

* Mogli smo, eli to nećemo da učinimo Ovekvom izmenom bismo, doduše, uštedeli dva bajte ali tade niko ne bi znao šta tekva naredba radi e de ne računemo što bi, ćak i opremljen mapom sistamskih promenljivih pomislio de se nalazi ne pogrešnom putu, jar ni jedne sisiemska promenljiva ne počinje od 10887. Ovako je svakome ko se upozneo sa meiim tajnema' naredbe PTR sve sasvim jasno

Program sa slike 2 ilustruje ono što smo do sade uradili Treba obratiti pažnju de se program ovoga pute ne izvršava neprakidno već se zehvaljujući naredbi broj 80, postavlje samo 20 problama Posle prikazivanje poslednjeg galeksije nailazi na naredbu 260 koju naprekidno izvršave ne kvareći teko izgled tabela Rezultate koje je postigio dete jasno, ne može da „uništi", bar ne po previlima „fer-pleja"

-- ??????

* Pod fer-pisjom na podrazumevamo način koga bi se setiio ćak i dete koje ja delo 20 pogrešnih odgovora – iskjučivenja računara iz napajanja Osim toga pritisak na BRK prekide izvršavanja programe Ćek i eko pretpostevimo de ovaj taster nije neročito upedljiv taster se RESET na zednjoj strani prosto memi de buda pritisnut

Tsstrejmo program

— Pokušajmo najpre da demo tečan odgovor Dobro je kaže BRAVO A pogrešan? i to je u redu Da sačekamo više od 25 sekundi Nikade 25 sekundi nije ovoiiko trajeio Prošao je i minut i po i još ništa izgiede da program ne radli

* Pokušajmo sade da demo tačan odgovor Evo kaže da smo predupo razmišljali!

— U pravo vrame Zašto to nije ispisao renije?

Zato što mu nismo rekli de to uredi Korietili smo naredbu INPUT koje nalaze računeru de čeka sve dok korisnik na otkuca treženi podatek i da onda nastevi sa redom Vrame u međuvremenu teče što interni časovnik iepo registruje ali to ni na koji način ne možemo de registrujemo dok računar ne nastevi sa radom Pre nego što počnemo da radimo na ovom problemu traba da primetimo još jeden pokušajmo da sačekamo minut i deset sekundi

-- Računer kaže de je odgovor dobar ieko je vreme prekoračeno više nego dvostrukoli!

* Problem je u tome što smo ispitujući vrame, testirali samo sekunde Kade minut proda sakunde ponovo počinju od nule Mogli bismo, jasno da provaravamo i minute pa i časove. eko hočemo de budemo sasvim dosledni eli se ovim nećemo baviti Ovaj problem će sa rešiti sam od sebe kada rešimo onaj mnopo veći -- keko de registrujamo sam trenutak kada vrame istekna O tome nerevno siedećeg časa

Čas 100 — Naredba KEY

Od naredbe INPUT, na žaiost, moramo de odustanemo jer ne možemo da je promenimo tako da vrača kontroiu našem programu de bismo kontrolisali vreme

— Naredba INPUT je previše zgodna! Zar problem kontrole vremena ne može baš nikako da se reši?

* Može, ali ne na nivou kojim se bavimo (iskusnijim korisnicima koji čitaju ovu školu a žele da provere svoja znanja o interapiu predlažemo da pokušaju da sa-slave odgovarajući mašinski program za modifikovano INPUT). Moraćemo kao što rekosmo da sa odreknemo INPUT-a i proučimo jedinu preostalu naredbu za testiranje tastature — naredbu KEY. Ona ima veliku prednost nad naredbom INPUT utoliko što se izvršava samo jedan tranutak — kada naiđe na naredbu IF KEY (1) GOTO 1000 računar proverava da li je taster A pritisnut i ako jeste ide na liniju 1000, a ako nije na sledeću čekanja ni u kom slučaju, nema

Moramo pre svega da razmotrimo koju grupu tastera treba testirati. Deta- po prirodi stvari treba da otkuca neki broj što znači da tastere sa slovima i specijalnim znacima možemo sasvim lepo da ignorišemo. Ostaje da treba testirati samo tastere 0–9 odnosno prema tabeli sa strane 21 upustva KEY (32) — KEY (41). Njih ćemo najbolje da testiramo u petlji i to jedan po jedan na primer ovako

```
10 X$=""
20 FOR I=32 TO 41
30 IF KEY (I) ELSE NEXT I  GOTO 20
40 X$=X$+CHR$ (I+16)  GOTO 20
```

Najteže je shvatiti liniju 40. Na početku smo promenljivoj X$ dodelili vrednost praznog stringa tj. alfanumerika koji ne sadrži nijedan simbol. Čim otkrijemo da je dete pritisnulo neko slovo treba da ga pridružimo uz taj string koji će tako iz trenutka u trenutak postajati sve duži. Ovo dodavanje slova je zapravo sabiranje alfanumerika. Ako na primer X$ ima vrednost GALAKSIJA" Y$ vrednost „JA" a mi izvršimo X$=X$+Y$, PRINTX$ će dati reč GALAKSIJA koja je dobijena nadovezivanjem JA na GALAKSI

Nadovezivanjem alfanumerika 0 na X$ bi mogli da se realizuje sa X$=X$+_0 ili prema tabeli sa strane 18 X$=X$+CHR$ (48). Primetimo da za testiranje tastera na kome je nula koristimo IF KEY (32) i da je 48=32+16. U ciklusu koji smo maločas dali promenljiva I ima ulogu brojača koji sputuje tastere koji su pritisnuti. Čim naiđe na neki od njih galaksija na promenljivu X$ nadovezuje CHR$ (I+16) što predstavlja upravo ASCII kod pritisnutog tastera. Možda neće biti loše da ovo sami isprobate na nekoliko primera i uverite se da program korektno radi. Da biste međutim ovo uopšte mogli da uradite treba da otkrijete trenutak u kome je korisnik završio sa kucanjem broja. Signal za to je naravno pritisak na ENTER, pa ćemo program dopuniti naredbama

```
30 IF KEY (I) ELSE NEXT I  IF KEY (48)
GOTO 100 ELSE GOTO 20
100 PRINT X$  GOTO 10
```

Startovaćemo program u kucati proizvoljne brojeve. Po pritisku na ENTER računar će prikazati broj koji smo otkucali i početi

— Zašto ovaj program treba da bude ovoliko temeljan'? Da bi neka cifra bila registrovana treba pritisnuti neki taster i poduže ga držati pritisnutim. Sa druge strane ovu dugo pritiskanje može da ima i nepriJatnu posledicu da računar neki taster

```
10 '    TABLICA MNOZENJA
20 '
30 T=0
40 N=0
50 P=0
60 HOME
70 PRINT AT 256,"TACNI";AT 265,"NETACNI";AT 276,"BEZ ODGOVORA"
80 FOR L=1 TO 20
90 PRINT AT320,T;AT 331,N;AT 345,P
100 PRINT AT 0,"KOLIKO JE'
110 A=INT(RND*99+1) B=INT(RND*99+1) C=A*B PRINT
120 PRINT AT 9,A," *",B,"?'
130 M=0 X$="" Y$="00 00 00" DOT*
140 IF KEY(48)*(M>0) GOTO 270
150 IF KEY(29)*(M>0) GOTO 330
160 FOR I=32 TO 41
170 IF KEY(I) ELSE NEXT I GOTO 200
180 X$=X$+CHR$(I+16) M=M+1 IF M>10 GOTO 220
190 PRINT AT 19,X$
195 NEXT I
200 S=VAL(PTR Y$+6)
210 IF S<25 GOTO 140
220 PRINT AT 0,"PREDUGO RAZMISLJAS"
230 P=P+1
240 PRINT AT 32,A," *",B," =",C
250 FOR I=1 TO 500 NEXT I NEXT L
260 GOTO 260
270 IF VAL(PTRX$)=C GOTO 300
280 N=N+1 PRINT AT 0,"NIJE TACNO!"
290 GOTO 240
300 PRINT AT 0,"B R A V O !"
310 T=T+1
320 GOTO 250
330 M=M-1
340 BYTE PTRX$+M,0
350 PRINT AT 19,X$
360 GOTO 200
```

registruje dva puta a mogućnost da stvar korigujemo pritiskom na ← izgleda ne postoji

■ Problem je u tome što galaksija nije mnogo brz računar a bajzik nije mnogo brz jezik. Dok Galaksija ne prođe kroz čitav ciklus i ne naiđe na pritisnut taster može da prođe dosta vremena. Rešenje ovoga problema ne postoji – mogao bi jedino da se napiše poseban program na mašincu koji kao što rekosmo izlazi iz okvira ove Škole. Što se tiče korigovanja — tu pomoći ima. Treba, najpre da testiramo taster – (KEY (29)). Ako je on pritisnut, treba da promenljivoj X$ oduzmemo jedno slovo što i nije tako jednostavno kako na prvi pogled izgleda. Treba pre svega da naučimo da galaksija smešta alfanumerike u standardnom ASCII zapisu pri čemu se iza poslednjeg slova nalazi 00 koja označava njegov kraj. Cilj je da lu nulu pomerimo za jedno mesto ulevo

Da pristupimo memorIji treba nam, jsano naredba BYTE i promenljiva koja će nam govoriti koliko smo slova do tog momenta otkucali. To će biti promenljiva M koja će na početku dobiti vrednost 0 i koja će zatim biti povećavana kad god na X$ nadovežemo neko slovo. Kada poželimo da odsečemo poslednje slovo izvršićemo M=M–1 BYTE PTRX$+M 0 i tako skratili X$

— Ali ovom naredbom mi nismo pomerili nulu za jedno mesto ulevo! Jedno stavno smo na lo mesto upisali nulu ali se ona i dalje nalazi i na sledećem mestu – kraj stringa tako označavaju dva nula bajta!

■ Tačno ali to ovde nije mnogo bitno. Prvi nula bajt je za galaksiju oznaka kraja stringa dok drugi ni u kojoj prilici ne biva tretiran jer jednostavno nije deo korisnog sadržaja promenljive X$

Na trećoj slici je prikazan program koji ćemo nazvati konačnim. U njemu ima još nekoliko mesta koja vrede prokomentarisati. Promenljivu M smo osim za redaktoru upotrebili za neke druge slučaje galaksija najpre proverava da li je otkucani string duži od deset slova. Ukoliko bi se dale igralo kucajući broj duži od 16 slova (po prirodi stvari besmislen kao rezultat množenja dva dvocifrena broja) kolika je maksimalna dužina galaksijinoj alfanumerika mogli bi da nastupe razni problemi. Uz ovaj test računar će svaku ovakvu igru proglasiti za predugo razmišljanje i preći na sledeći problem

Testirano je osim toga da li je taster ENTER ili — pritisnut pre nego što je otkucan neki broj. Ukoliko jeste pritisak ovog tastera biva ignorisan jer bi mogao da izazove nove probleme. To smo učinili logičkim množenjem iskaza koje je opisano u 22 poglavlju Uputstva za upotrebu galaksije (strane 22-23)

I poslednje pitanje

— Ovih sto lekcija je nekako mnogo brzo prošlo. Čini mi se da je njihovo numerisanje bilo u najmanju ruku čudno

■ Lekcija nije bilo sto nego 100. A binarni sistem programeri često koriste

Dejan Ristanović

Spectrum

Kako postići besmrtnost

Program	Adresa	Sadržaj	Posledica
COOKIE	28698	0	bezbroj života
PSSST	24984	0	bezbroj života
Manic Miner	35136	0	bezbroj života
	34269	n	n= broj života, n < = 32
Pyramid	44685	0	neograničena energija
Arcadia	25776	0	bezbroj života
Hunchback	26888	0	bezbroj života
Transversion	26020	0	bezbroj života
Jet Set Willy	34483	195	eliminisanje zaštite
	35899	0	bezbroj života
ZIP-ZAP	53750	0	bezbroj života
	53751	0	bezbroj života
	53752	0	bezbroj života
	53753	0	bezbroj života
	54141	0	bezbroj života
	54142	0	bezbroj života
	54143	0	bezbroj života
	54144	0	bezbroj života
	54036	0	bezbroj života

Sva video igre, na ovaj ili onaj način, ograničaveju broj života glavnog junaka Često je to i istovremeno ograničavanje raspoložive količine energije kiseonika hrane ili vremena Posledicu znamo igrač veliku većinu vremena provodi u prvih nakoliko nivoa (koji mu ubrzo postaju nezanimljivi) Tek što uspe da pređe u sledeću (najčešće napoznatu i težu) fazu igre igreč izgubi nekoliko života, ne stigavši da uoči sva prepreka i razradi novu tektiku

(Istini za volju neke video igre kao šio je 'Penetretor' porad uobičajenog mode za igru pružeju i mogućnost uvažbavanja (training mode) igrača — igreč uvažbava pojedina teze igre ili jednostevno bira samo ono što voli)

Naravno bez ovih otežavajućih okolnosti većina igare bi izgubile svaku draž Ipak da bi isprevili ovu 'nepravdu' vredi ljubitelji igare su upornim redom otkrili ključne lokacije u programu koje je potrebno izmeniti da bi igrač postao besmrtan Teko oduzimaju mnogo vremena ovekvi poduhvati omogućuju slicanje novih znanje iz oblesili programirenja

Za ovu priliku smo odebreli nekoliko popularnih profesionalnih igara o čijim višim nivoima mnogi iscrpljeni spektrumaši još uvek mešaju Naveščamo lokacije koje je potrebno izmeniti neredbom POKE naposredno pre startovanja programa Većinu ovih lokacije je otkrio Eliša Kabiljo iz Beograde Za program Pyremid zastužan je Zoran Štrumbergar e za program Arcadia Slobodan Vujnović oboica iz Zemuna

Iz tabele se vidi de Manic miner može da se prekroji i tako da igreč ima tiksan broj života ali na veći od 32, jer se u tom slučaju dogadaju čudna stvari ne ekrenu Na primer eko želite de igrate sa deset života kucajte POKE 34269 10

Unošenje ovih izmena (u žargonu) petch, zakrpa) vrši se na sledeći način

1 Učita sa bejzik program koji učitave ekren i mašinski deo, iii delove programa To se postiže komandom MERGE
2 Pronalazi se linije u kojoj se izvršava naredbe RANDOMIZE USR xxxxx (ili PRINT USR xxxxx ili nešio slično) Ovom nardbom bejzik nepovreino skače u mašinaki program ne edresu xxxxx, pa se izmane moreju unati pr njenog izvršenje
3 Liniju ćamo Izmeniti teko što ćemo naposredno pre ove naredbe ukucati POKE edrese sadržaj iz tabele
4 Sada možemo de kažemo RUN ili GOTO da bi se nastavilo normalno učitavenje ostelih delova programa Još je prektičnije sačuvati ovaj izmenjeni program sa ne primer SAVE „NOVI" LINE i pomoću njega kasnije vršiti učitavanje programa

Za izmenu programa Jumping Jack potrebno je ubaciti tri nova linije i program sačuvati se SAVE „JJ-ET" LINE 10

```
10 POKE 26034 0 POKE 26035 91 RESTORE 30 FOR a=23296 TO 23309
20 READ d POKE e d NEXT e GO TO 2
30 DATA 33 0 0 34 141 38 24 34 143,117,195 66 111
```

Nedavno se u literaturi pojavilo još nekoliko zahvete koji omogućavaju ulazak u bilo koji nivo Manic Minere Postupak je prilično čudan učiteti normalno program, zatim pritisnuti ENTER i sledeći niz brojava: 6031769 Na dnu ekana bi trebalo da se pojevi cipele Pritiskajući ISTOVREMENO brojeve iz tabele koja sledi, ulazimo u odgovarejući nivo igre

nivo	Kombinacija
1	6
2	61
3	62
4	621
5	63
6	631
7	632
8	6321
9	64
10	641
11	642
12	6421
13	643
14	6431
15	6432
16	64321
17	65
18	651
19	652
20	6521

S Vujnović

Spectrum

64 znaka u redu

Ovaj program vam omogućeva da na iečunaru ZX Spectrum ostvarile ispis tekste na ekranu u širini od 64 slove u jednom redu Program je u potpunosti uradan u bejzniku Zbog ograničenja koja nameće rezolucija na ekrenu slove eu nešto manje čitljiva nego originelna, eti uz malo podešavanje osvatljenja i koniresta na vašem TV eparetu to na bi trebelo de predstevije problem

Da bi se razumelo kako program redi potrebno je upoznati se sa znečenjem sistemske varijebile CHARS Te varijabie stoji na edresi 23606 i 23607 i pokazuje Spectrumovom operativnom sistemu gde trebe da traži definiciju oblika slova koje ispisuje Prvobitna vrednost za CHARS je 23606=0 i 23607=60 i ona pokazuje na ROM (tikanu memoriju) pa etoga i dobijemo slova odmah po uključenju računare Ta sistemska varijebile se može postaviti na bilo koju drugu vrednost u okviru RAM-a Ako se na novoj adresi nelazi neka druga nova definicije slova (na primer ćirilica) Spectrum će je prihvatiti kao svoju sopstvenu

Da bi ta osobina mogle da se iskoristi za dobijenje 64 slova u redu na ekranu potrebno je definisati dva nova seie znakova Oba ova seta će sadržavati ista slova samo što će se u jednom setu slovo nalazili u levoj polovini Spectrumovog standardnog znaka, e u drugom setu u desnoj polovini Sada je još potrebno de navedemo računar de ta dva znaka odštampa u jednoj standardnoj pozicilji na ekranu, što se postiže naredbom OVER 1, koje, inače, Ima zadatak da preštampa dva elova jedno preko drugog

Niz znakove može Imati do 64 slova i smešta se u niz B$, koji može biti dimenzionisan ne novu širinu ekrana Te dimenzije ne može biti veća od 64 i ne manja od 1 Program će prihvatiti niz slova bilo koje dužina (do 64)

Potrebno je takođe objasniti funkciju promenljive F Ona siuzi programu de za evako perno slovo zedrži poziciju za pisanje na ekranu i teko omogući de, preko OVER 1, budu napisane dve slova umesto jednog u jednoj sienderdnoj slovnoj poziciji

Ako vam kojim slučajem program krehire sa novim setom znakova, povratak ne stenderdni set se vrši sa GO TO 9999

Program omogućava štampanje na bilo kom mestu na ekranu i to na način koji je sličan Spectrumovoj PRINT AT instrukciji To znači da pre poziva podprograme ireba posieviti dve promenljive „r za red i „p za kolonu u kojoj želite da štampate Takođe treba biti obazriv da dužina teksta koji štempeie na prede krajnju desnu kolonu jer ća se tade jeviii greška

Sam program je veoma kietak i obuhvate samo linije 9000 do 9999 Ceo potprogrem se nalazi u liniji 9000 Linije 9988 vrši eutomatsko enimanje potprograma I novog sete znakova Linije 9990 je za eutometsko učitavanje sete znakova e 9999 vam pomaže da se vratite u Spectrumovu definiciju znakova Sama definicije dva nova sata znakova je prilično obimna i zehieva 1540 bajtova na kraju memorije (iznad RAMTOP-a)

Uz knjige koje su alat za stabilizaciju

MOŽETE ZNAČAJNO POVEĆATI VAŠU PRODUK-TIVNOST

Izdala je iz štampe knjige

D.S. Arandelović
1. KOMPJUTERSKA ORGANIZACIJA U SAVREMENOJ PRIVREDI

D.S. Arandelović, dipl. inž. stručnjak za organizaciju rada, naučni radnik, profesor, savetnik UN i vši konsultant H B Maynard INC: Management Consultants (USA). Autor 14 knjiga iz oblasti organizacije, industrijskog inženjeringa i tehnike. Glavni redaktor „Privredni pregled" i „Knjiga Komerc". Knjiga Kompjuterska organizacija i savremena ekonomika biblioteka i biblioteka velikih priručnika „Privredni pregled" postupaka koji se primenjuju u savremenim uslovima. Pored „ručnih" moderne metode organizacije i rada u privredi obrađuje i sređivanje podataka koji se primenjuju u savremenim automatizacije poslovanja kojima se postiže visoka produktivnost, ekonomičnost i konkurentnost.

Narodnu su obrađene metode upravljanja procesa proizvodnje i tržištu. Obrađeni su primeri kompjuterskih sistema koji se primenjuju u savremenim preciznih normativa (MTM). Obrađeni su primeri kompjuterskih sistema izvršenih sistemskih rešenja kompjuterskih sistema koji se primenjuju u savremenim kojima se može uspešno vanja problema, iskustva, nesporazume i tekove u morati se studirati pri modernizovanju i radnih organizacija.

Nekompjuteraku organizaciju sprovedite tako da je možete u svako doba prevesti u kompjuterski sistem

Strana 332, format 18×25 cm, povez tvrd sa zaštitnom višebojnom omotnicom, latinica. **Cena 1.100,– dinara**

Frederick W. Hornbruch
2. POVEĆANJE PRODUKTIVNOSTI

Deset praktičnih primera kazuju kako je u deset američkih preduzeća da povećana produktivnost 25—50% pa čak i 100% a organizovanim rekorde prozvodnje i rekordnog dohotka.

Data je analiza sledećih iskustvenih primera.

a) Fabrika mašina, povišila produktivnost 75%; i povećala produkciju za 250%. b) fabrika metalnih proizvoda, povišla produktivnost za 63%. c) Državni statistički biro 52% d) Bendiks, radna zajednica 68—94% e) Velika radionica za održavanje i opravku fabričkih postrojenja 46% f) izdavačko preduzeće 25% g) Fabrika Multi-filj 90% i) Banka značajno povišla produktivnost.

Primeri iz ove knjige vam ne samo za 5 ili 10% nego za 50—100%

Povećajte produktivnost ne samo za sebe već i za svoje saradnike.

Strana 346, format 18×25 cm povez tvrd, višebojna omotnica. Latinica. **Cena 750,– din.**

H.B. Maynard
3. SAVREMENA ORGANIZACIJA PROIZVODNJE

Prirućnik modernog upravljanja proizvodnjom

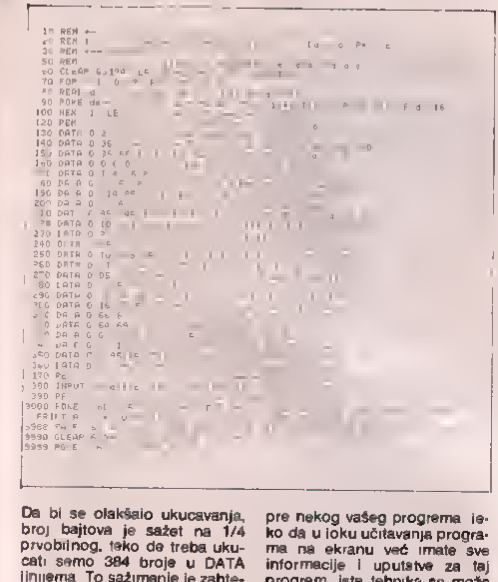
Jedno od najznačajnijih svetskih izdanja, na kome je sarađivalo 85 svetskih naučnika, stručnjaka i privrednika u redakciji najvećeg živog priručnika je namenjen savremenim rukovodiocima proizvodnje, kao pomoć u rešavanju

BIBLIOTEKA PROGRAMA

kasefu. Od sada pe ne dalje možete ga učitaveti sa LOAD 64 slova/1' ili samo LOAD"

Osnovna vrednost ovog programa je što umesto 704 slovna meste (ako isključimo donje dve sistemske linije) omogućava korišćenje 1408 slovnih mesta ne ekranu što znači i dva puta više informacije u isto vreme

Jedan predlog bi bio da se tako formirani ekran smesti keo SCREEN$ datoteke na traku a



Da bi se olakšalo ukucavanja, broj bajtova je sažet na 1/4 prvobitnog, tako de treba ukucati samo 384 broje u DATA linijema To sažimanje je zahtevalo jednu malu rutinu koje će izvršiti potrebne transformacije i premestiti karakter set na njegovo pravo mesto Rutina za premeštenje sete znakove je u DATA linjama 130 do 360 Linije 380 služi ze demonsteciju rede programa

Posle ukucavanja programa i njegovog stertovanja, mogu se izbrisatr sve DATA linije kao i rutina ze premeštanje sete znekove

Pošto ukucate ceo program oikucajte RUN da izvršile definisanje novog sete znakove Program će tade izvršiti malu demonstraciju rede Zatim otkucajte GO TO 9999 i izbrišite linije 10 do 390 jer vam više nisu potrebne Sa GO TO 9988 izvršite snimanje podprograma i definisanog sete znekove na

pre nekog vašeg programa ieko da u loku učitavanja programa na ekranu već imate sve informacije i uputstve za taj progrem ista tehnika se može iskoristili de smestite sadržej treke na njen početak Učiteva nje detoteke SCREEN$ traje oko 40 sekundi- Dovoljno je, znači, da toilko ostavite na početku svake treke pa de kasnije tu snimite njen sadržaj koji ste napisali uz pomoć ovog programa Datoteka tipa SCREEN$ se učitava sa LOAD "" CODE

Borivoj Perzić, dipl ing

„galaksija" master-majnd

Maslemajnd je poznate logičke igre na epacijalnoj tabli se raznobojnim pribadačama Jedan od igrača sa zednja strene tebsie postavija četiri proizvoljne pribadače u proizvoljnom poretku, postoji šest boja pribadeča e naki od otvora mogu da ostanu i prazni (sedma boja)

Zadatek drugog igrača je da pogodi zadatu kombinaciju U tom cilju on u prvi red tebie postevlja četiri pribadače dok drugi igreč saopštava koliko se ta kombinacija siže sa onom koju je on zamislio Za svaku pribadaču koje je postevljena ne dobrom mestu u dobroj boji igreču se deje jedna crna e za svaku koja je pogođena po boi ali ne i po mestu — jedne bele pribadača (crne i bele pribadeče na mogu da se nadu u zadatoj kombinaciji, one su samo informativne) Na osnovu ovih podeteke igreč pokušava da, u nekom od sledečih pokušaje, dobije četiri crna pribadeče — znak da je pogodio kombinaciju

Pošto „galeksija" nema kolor grafiku boje su zamenjena brojevima 0—6 Da bi igre bila zanimljivija igreč i računar paralelno pokušavaju da pogode kombinaciju koju je protivnik zamislio — onaj koma to pre uspe dobije jedan poen

Nejpre treba sa NEW 512, rezervisati prostor za mašinski program i unati ga pomoću UTM-a ili nakog drugog hex-loedera Zatim treba ponovo otkucati NEW 512 i uneti bejzik program Snimičemo ava ovo

```
1 '
2 ' "VENTILATOR" 1 "GALAKSIJA"
3 '
4 '          7.4.1984.
5 '
100 W=RND:FOR I=0 TO 3
110 BYTE &2E80+I,INT(RND*6)
120 A(I)=PEEK(&7FFF+I)
130 NEXT I
135 HOME
140 PRINT "JA SAM RACUNAR GALAK
SIJA":P."A TI":I= X#
145 HOME
150 PRINT X#;AT 76;"GALAKSIJA"
200 P=96
210 IF WK0.5 G=5#0
220 PRINT ATP;:INPUT Y$
230 FOR I=0 TO 3
240 BYTE&2C3A+I,A(I)
250 D=BYTE(PTRY$+I)-48
260 IF (D<0)+(D>6) P 220
270 BYTE &2C42+I,D
280 NEXT I
290 D=USR(&2C46)
300 PRINT ATP+24;I.Y$
310 IF D=4 0.1000
320 D=USR(&2C60)
330 P.P1
340 FOA I=0 TO 3
350 BYTE &2880+P+76+I,BYTE(A+II
+48
360 NEXT I
540 PRINT ATP+24;I.Y$
550 D=BYTE(PTRY$+I)-48
560 IF (D<0)+(D>6)AT G.5#0
570 IF D=4 0.2000
580 BYTE A+4,0
590 D=BYTE(PTRY$+I)-48
600 IF (D<0)+(D>6)AT 0.5#0
200 BYTE A+5,D
710 A=USR(&2C8A)
711 IF A=&FFFF 0-3000
720 P=P+32:P 220
7000 R=R+I
1000 PRINT AT 362,"E A A V D"
1020 PRINTi:PRINT X#,AT 430,R
1030 PRINT "GALAKSIJA",AT 462,K
1040 PRINT AT 499,"PRITISNI RET
"
1050 IF KEY(49) EL60 0.1050
1060 IF KEY(49)T 0.1060
1070 FOA I=0 TO 3
1080 BYTE &2E80+I,INT(RND*6)
1090 A(I)=INT(RND*6)
1090 NEXT I
1100 A=USR(&2C80)
1110 W=1-W:0.705
2000 K=K+1
2010 PRINT AT 362,"POBEDIO SAM!
"
2020 GOTO 1020
3000 PRINT AT 362,"V A A A B !"
3010 K=K+1
3020 0.1020
4000 HOME
4010 PRINT "          * MASTERMAJN
D  *"#
4020 PRINT
4030 PRINT "   PRAVILA IGRE SU,
NADAMO SE,"
4040 PRINT "OPSTE POZNATA. OVAJ
PROGRAMT"
4050 PRINT "OMOGUCAVA PARALELNU
IGRU PRO-"
4060 PRINT "TIV RACUNARA. MOZE
S LI DA PO-"
4070 PRINT "BEDIS?";P.
4080 PRINT "   IGRAJU CIFRE 0-6
NA CETIRI"
4090 PRINT "MERTA. SVI POGACI S
E KUCAJU"
4100 PRINT "NORMALNO IKOMANDA I
NPUTI"
4110 PRINT AT 499,"PRITISNI RET
"I
4120 IF KEY(49) EL60 0.4120
4130 IF KEY(49)T 0.4130
4135 HOME
4140 RET
```

```
&2C3A: B2 00 B5 05 FF FF 02 FF
&2C42: 07 02 FE FE 21 3A 2C 11
&2C4A: 3E 2C CD 77 2C CD 7D 2C
&2C52: CD 86 2C 23 13 7B FE 44
&2C5A: 20 F6 2A 00 68 CD 7D 2C
&2C62: 2C 11 42 2C CD 86 2C 13
&2C6A: 7B FE 46 20 F7 23 70 FE
&2C72: 42 20 EE 78 E5 07 00 00
&2C7A: ED B8 C9 06 00 2F 3E 2C
&2C82: 11 42 2C C9 7E E5 4E EE
&2C8A: E9 C0 04 3E FF 77 E5 D5
&2C92: E1 3E FE 77 E1 C9 77 E1
&2C9A: C9 00 00 00 00 00 00 2A
&2CA2: 86 2E 53 09 00 00 00 00
&2CAA: 00 00 00 00 00 00 2F 00
&2CB2: 2E 22 00 2C 2A 00 2C C9
&2CBA: 2A A0 2C 11 A6 00 19 E5
&2CC2: D9 DF 2A 00 2C 11 04 00
&2CCA: ED 00 EB 36 00 D9 22 00
&2CD2: 2C 2F 52 09 22 A4 2C 2A
&2CDA: A4 2C 2B 7C B5 22 A4 2C
&2CE2: 21 FF FF C0 2A A0 2C 3A
&2CEA: 7E FE 07 20 05 36 00 23
&2CF2: 18 F5 2F 00 2E 22 A2 2C
&2CFA: 11 3A 2C 2A 00 2C CD 77
&2D02: 2C 11 42 2C 2A A2 2C C0
&2D0A: 77 2C CD 46 2C 2A A2 2C
&2D12: 11 04 00 19 7E 00 28 3F
&2D1A: 23 E5 CD 60 2C E1 7E 00
&2D22: 20 B5 23 22 A2 2C CD AC
&2D2A: 2D 2A 15 CD AC 2D 21 FF
&2D32: F2 29 11 A0 2D 06 0C CD
&2D3A: AC 2D 3A 33 20 E6 07 20
&2D42: F9 2F FF 29 36 20 2B C9
&2D4A: 3C 20 F9 C3 00 2C 4E 42
&2D52: 54 55 46 53 4E 42 40 4F
&2D5A: 45 45 46 40 42 4F 21 53
&2D62: 4A 54 55 42 4F 50 57 4A
&2D6A: 50 2F 32 3A 39 35 5F 53
&2D72: 4A 55 4A 54 4F 4A 21 45
&2D7A: 46 40 7A 3D 77 05 23 73
&2D82: 20 F8 C9 3E 73 3A 2A 23
&2D8A: 3D 20 FA C9 00 00 00 00
&2D92: 00 00 00 00 00 00 00 00
&2D9A: 00 00 00 00 00 00 00 00
&2DA2: 00 00 00 00 00 00 00 00
&2DAA: 00 00 00 00 00 00 00 00
&2DB2: 00 00 00 00 00 00 00 00
&2DBA: 00 00 00 00 00 00 00 00
&2DC2: 00 00 00 00 00 00 00 00
&2DCA: 00 00 00 00 00 00 00 00
&2DD2: 00 00 00 00 00 00 00 00
&2DDA: 00 00 00 00 00 00 00 00
&2DE2: 00 00 00 00 00 00 00 00
&2DEA: 00 00 00 00 00 00 00 00
&2DF2: 00 00 00 00 00 00 00 00
&2DFA: 00 00 00 00 00 00 00 05
&2E02: 07 05 01 02 07 00 02 05
&2E0A: 00 01 02 02 01 07 07 07
&2E12: 03 02 00 02 07 07 04 03
&2E1A: 01 02 00 00 07 02 04 03
&2E22: 00 00 00 00 00 00 00 00
&2E2A: 00 00 00 00 00 00 00 00
&2E32: 00 00 00 00 00 00 00 00
&2E3A: 00 00
```

drugi način ukucavanja programa je da u loku učitavanja programa na ekranu već imate sve informacije i uputstve za taj program ista tehnika se može iskoristili de smestite sadržej treke na njen početak Učiteva nje detoteke SCREEN$ traje oko 40 sekundi- Dovoljno je, znači, da toilko ostavite na početku svake treke pa de kasnije tu snimite njen sadržaj koji ste napisali uz pomoć ovog programa Datoteka tipa SCREEN$ se učitava sa LOAD "" CODE

BIBLIOTEKA PROGRAMA

ne kasetu sa SAVE e zatim stertovati program sa RUN Posle zagievlja računer će zatražiti da se predstavimo e zatim ča početi igre U toku predstavljanja treba de zamisilmo broj koji će računer pogedati; to je kako rekosmo, četvorocifreni broj, u koma nijedne citra ne ema de bude veća od 6 dok se citre nerevno, emaju ponavljeti Kade je na nas red, treba de otkucamo pretpostevku (npr 1234) i pritisnemo RETURN (ENTER) i računer će prikazati kolliko bismo crnih odnosno belih pribadača dobili (21 označave dve cine i jednu balu) e zatim će on pokušati da pogodi našu kombineciju Odgovaramo mu na isti način — kucajući 21 03 ili neki sličan broj i pritiskajuć ENTER igra se nastavije sve dok neko ne pogodi protivničku kombineciju (skor 40) računar tade ispisuje trenutni skor i igre se nastevlja.

Dejan Ristanović

„galaksija“ Evolucija

Evolucije (Life) je poznate emulecije živote Na nekom steništu korisnik zadaje početni organizem koji se sestoji od određenog broja živih ćelije dok su preostala polja mrtva Taj organizem se razvije u strogim evoluciorim uslovima — usamljene ćelije i ćalije koje žive u preneseljenim područjima brzo umiru dok se u umereno naseljenim predelima iz časa u čas rađaju nove ćelije Neki početni organizmi, posle određenog broja generacije umiru dok drugi prelaze u neke stabilna forme Za mnoge početne organizma se verovati ili ne, posle određenog broja generacije javlja struktura koja čak i baz mnogo maštе podseća ne čoveka

```
 10 '
 20 "VENTILATOR" 1 "GALAKSIJA"
 30 '
 40 '          31.3.1984.      '
 50 '
 60 '
 70 HOME
 80 P AT165,"**********************
**"
 85 P AT197,"*                     
*"
 90 P AT229,"*      EVOLUCIJA      
*"
 95 P AT261,"*                     
*"
 99 P AT293,"**********************
**"
100 FOR I=1 TO 2000:N I
110 HOME
120 PRINT "              KOMANDE:"
130 PRINT
140 P
141 PRINT "STRELICE          ZA PO
KRETANJE"
142 P
150 PRINT "SHIFT STRELICE ZA BA
ZU"
160 PRINT "                  KRETA
NJE
165 P
170 PRINT "          ENTER  ZA OS
TAVLJANJE"
180 PRINT "                  CE
LIJE"
190 PRINT "          SPACE  ZA BA
USANJE"
195 P.
200 PRINT "              L   ZA PO
CETAK"
1000 A=USR(&2C3A)
```

```
&2C3A: 1E 18 21 C8 29 01 2E 20
&2C42: 0A 3D 77 23 03 1D 28 F8
&2C4A: 01 46 2D CD 1B 2D 01 4D
&2C52: 2D CD 1B 2D 3A 38 20 E6
&2C5A: 01 28 ED 3A 39 20 E6 01
&2C62: 26 F9 21 FF 29 36 20 2B
&2C6A: 7C FE 27 20 F8 01 01 00
&2C72: 38 28 21 00 28 CD 11 2D
&2C7A: 38 20 21 E0 2F CD 11 2D
&2C82: CD 11 2D 21 21 28 16 20
&2C9A: 36 20 3A 0C 20 E6 01 CA
&2CA2: 54 2D 3A 38 20 E6 01 20
&2CAA: 02 16 2A 3A 1F 20 E6 01
&2CB2: 20 02 16 20 3A 1E 20 E6
&2CBA: 01 20 06 01 ED FF CD EC
&2CC2: 2C 3A 1C 20 E6 01 20 06
&2CCA: 01 20 00 CD EC 2C 3A 1D
&2CD2: 20 E6 01 20 06 01 FF FF
&2CDA: CD EC 2C 3A 1E 20 E6 01
&2CE2: 20 00 01 01 00 CD EC 2C
&2CEA: 18 00 E5 09 7E FE FF 28
&2CF2: 02 E1 C9 7A D9 E1 77 D9
&2CFA: 5A 36 29 01 00 0A 3A 3C
&2D02: 28 E6 01 28 03 01 00 1A
&2D0A: 00 78 FE 00 C8 18 F9 36
&2D12: FF 09 30 20 FA C9 21 F9
&2D1A: 29 1E 07 0A 77 23 03 1D
&2D22: 20 F9 21 00 9A 28 7C FE
&2D2A: 00 20 DD C9 45 4A 48 42
&2D32: 4F 21 21 53 4A 54 55 42
&2D3A: AF 50 57 4A 5D 21 21 52
&2D42: 3A 39 35 2F 3C 45 4E 54
&2D4A: 45 52 3E 20 20 20 20 20
&2D52: 20 28 72 3A 3E 28 E6 01
&2D5A: 28 F9 01 21 28 16 FF 0A
&2D62: FE 28 28 02 1A 00 21 DF
&2D6A: FF CD C9 2D 21 FF FF CD
&2D72: C9 2D 21 1F 00 CD C9 2D
&2D7A: 7A CB E1 3A CB 02 FE 02
&2D82: 28 06 FE 02 20 06 CB 4A
&2D92: 28 02 CB F2 72 03 21 DE
&2D92: 29 37 3F ED 42 28 00 0A
&2D9A: FE FF 28 02 03 03 C3 5F
&2DA2: 2D 21 20 28 23 7C FE 2A
&2DAA: 28 00 3A 31 20 E6 01 CA
&2DB2: 44 2C C3 55 2D 7E FE FF
&2DBA: 28 EA CB 77 28 05 3E 2A
&2DC2: 77 18 E1 3E 20 18 F9 09
&2DCA: CD D2 2D 23 CD D2 2D 23
&2DD2: 7E E6 BF FE 2A C0 14 C9
&2DDA: 00 00 00 00 00 00 00 00
&2DE2: 00 00 00 00 00 00 00 00
&2DEA: 00 00 00 00 00 00 00 00
&2DF2: 00 00 00 00 00 00 00 00
&2DFA: 00 00 00 00 00 00
```

Da bista otkucali ovej program treba najpre de rezervišete prostor za njegov mašinski deo — to činite tako što kucate NEW 454 Posle toga treba da unesete mašinski deo pomoću UTM-e ili nekog drugog programa Ponovo otkucajte NEW 454 da obrišete UTM, pa unesite bajzik koji jednostevno, štampe uputstve Sve to snimite ne kasetu sa SAVE Posle toga ćete program uvek sterloveti jednostevnim RUN i sledili uputstve koje računer ispisuje

Dejan Ristanović

BBC Electron Kako presnimiti program?

Acornovi računari imaju sistem snimanje programe i podeteka ne kasati koji slobodno možemo de nazovemo genijelno zemišljenim. Podaci se upisuju u blokovima od po 256 bajtova koji su jesno razdovjeni jedan od drugoge — ispred svekoga bloka se nelaze potrebne informecije koje obuhveteju i ime samog programa Zbog čega je ovo zgodno? Pre svega možemo da uzmemo sveku treku smestimo je u kasetoton i premotemo ne bilo koje mesto Ako otkucamo *CAT i pritisnemo RETURN e zatim i PLAY na kasetofonu za trenutak ćemo videti ne koji je program kasete pozicionirena i koliko smo deleko od njegovog početka (računer uz ima programa ispisuje i broj bloka na kojl smo naišli i to heksadekadno)

Osim imene u okviru zaglevlje svakog bloka nelazi se i druge informecije To je najpre adresa u memoriji od koje program počinje de se učitave ukoliko korisnik nije naiedno drukčije (označava se kao load address); zatim adresa od koje programi počinje da se izvršave ukoliko ga učitavamo sa "RUN (, excution eddress) broj bloka od početka i (eventuelno) bit koji govori o toma da se iadi o zadnjem programu u bloku

Najjednostevniji nečin da presnimimo program je da, najpra otkucamo "OPT 1,2 e zatim "LOAD Učitevanje će započeti potrajeti izvesno vreme e zatim će računer Iza imena programe ispisati njegovu dužinu u bajtovime kao i load i execution edresu programa Adresa od koje se program učiteva i izvršne edresa imaju po četiri umesto dva bajte što ne bi trebalo posebno da nes zabrine — oni koji kupe dodatni procesor sa 64 K (ili pola magabajte) RAM-a shvetiće de program može da se učitave i u ovej novi mamorijski prostor, koji ne neki način treba adresireti Drugo zepažanje će biti još jednostevnije i korisnije pošto znemo odekle program počinje, kolliko bajtove ima i koje mu je izvršne adresa moći ćemo de ga snimimo ne kasetu baz mnogo problema — dovoljno je da otkucamo "SAVE ,ima FFFFxxxx + nnnn FFFFeeee pri čemu **nnnn** označave dužinu programa u bajtovima **xxxx** adresu od koje se on učitave a **eeee** izvršnu adresu

Ne žaiost, ne mogu svi programi da se presnima na ovaj način — u cilju njihove zaštite adresa od koje sa ovi programi učitaveju je odebrane tako da po učitavenju, kontrole ne može de se vrati bajzik interpretoru Te programe je neophodno učitali na nako drugo mesto a zatim snimiti tako da kopije bude identična orginelu što možemo samo koristeći sistemski potprogram OSFILE

OSFILE se poziva sa JSR &FFDD ili, ukoliko je poziv iz bajzika CALL &FFDD Pre toga je neophodno da u X i Y registre smestite edresu kontrolnog bloka koji kontroliše operaciju upisa odnosno čitenja (u X se smešta bajt manje e u Y bajt veća težina edrese bloka) e u A kôd koji kontroliše operaciju koju treba izvršiti. Tačna strukture kontrolnog bloka je deta ne streni 455 uputstve ze upotrebu BBC B u ekumulator se stavija nula ukolliko želimo de upisujemo ne traku e &FF eko želimo da čitemo sa nje

Posle čitanja programa sa trake računar će u svoj redni prostor smestiti parametre programa Adresa od koje se program učitaveo će biti smeštene od &3BE do &3C1, izvršne edrase od &3C2 do &3C5, dužine u &3C6 (MSB) i &3C8 (LSB) e ime programa počavši od &3D2 pa do prvog nula bajte koji predstevlja terminator Treba primetiti da dužina nije uvek tačna — ako je njen bajt manja težine (LSB) nule, bajt veće težina (MSB) treba veštački povećati za jedan da bi se dobio tačen podetak o dužini Uz pomoć ovih podeteka (koji se ovde po prvi put jevno objevljuju) može da se pripremi kontroini blok za snimanje programa koji omogućave pravljenje kopije

Čitave ove teorije je sažete u program PREPIS koji objevljujemo Asemblirajte ga i snimite na traku (pre kucanja sors programa otkucajte PAGE=&3000 a posle asemblirenja — "SAVE ,PREPIS" E00 +&B0) Pro-

```
  0 REM
  1 REM            PROGRAM PREPIS
  2 REM
  3 REM            MASINAC   &E00
  4 REM            IME       &E71
  5 REM            BLOK      &E7C
  6 REM            BLOK ZA
  7 REM            PREPIS:   &F00
  8 REM
  9 REM       Dejan Ristanovic 1984
 10 REM
 11 REM
 12 REM
 100 FOR I=0 TO 3 STEP 3
 200 P%=&E00
 300 [OPTI
 400 .ENTRY_POINT
 500 JSR &FFE0
 600 .ENTRY2
 700 LDX #&7C
 800 LDY #&E
 900 LDA #&FF
1000 JSR &FFD8
1100 LDA #0
1200 STA &E86
1300 LDA #&F
1400 STA &E87
1500 LDX #0
1600 .CYCLE1
1700 LDA &380,X
1800 STA &E7D,X
1900 DEX
2000 BNE CYCLE1
2100 .CYCLE2
2200 LDA &302,X
2300 BEQ PREPARE
2400 STA &E71 X
2500 INX
2600 BNE CYCLE2
2700 .PREPARE
2800 LDA #13
2900 STA &E71 X
3000 CLC
3100 LDA &3C8
3200 STA &E8A
3300 BNE .OK
3400 INC &3C8
3500 .OK
3600 LDA &3C6
3700 AND #&F
3800 STA &E8B
3900 .SAVING
4000 JSR &FFE8
4100 CMP #13
4200 BEQ REINIC
4300 LDX #&7C
4400 LDY #&E
4500 LDA #0
4600 JSR &FFD8
4700 JMP SAVING
4800 EQUS "(C)D.Ristanovic 6"
4900 EQUB &SC*4*
5000 ]
5100 P%=&E9E
5200 [OPTI
5300 .REINIC
5400 LDA #&0
5500 STA &E71
5600 LDA #0
5700 STA &E7E
5800 STA &E8F
5900 STF &E80
6000 STA &E81
6100 LDA #&F
6200 STA &E7F
6300 JMP ENTRY2
6400 ]
6500 NEXT I
6600 ?&E71=&80
6700 I=&E7C
6800 READ A:IF A=9999 THEN 7400
6900 ?I=A:I=I+1
7000 GOTO 6800
7100 DATA &71,&E,0,&F 0,0 0,0,0,0
7200 DATA 0,&F 0,0,0 0,0,0
7300 DATA 9999
7400 CALL ENTRY_POINT
```

grem se aktivira sa "RUN PREPIS Po učitavanju pritiskate RETURN de učitete sledeći progrem sa treke e SPACE de u memoriju učitani progrem snimite na drugu kasetu Ovo snimanje možete de ponavijete proizvoljen broj puta — sve dok na napravite dovoljan broj kopija Zatim možete de pritisnete RETURN i učitete sledeći progrem

PREPIS na žaiost ima ograničeno dejstvo pomoću njega ne možeie da učitate programe Hopper Starship Command i još nekoliko drugih Osim toga program „The Hobbit" je toliko dugečak de ga PREPIS vrlo ieško prepisuje O načinu na koji se presnimaveju i najbolje zaključani (kade sa *LOAD pokušamo de pročitemo posebno zaštićen program dobićemo poruku Locked) dugački programi govoriće mo eko za to bude interesa u Računarima 3 Tade ćemo objeviti i poboljšanu verziju programa PREPIS koja se zove DCOPY

BBC/Electron
TAJM AUT

Toga dana sve vam je išlo savršeno — učitali ste evoju najomiljeniju igru i, iz prvog pokušaje, postigli svoj apsoiutni rekord — došli ste do 12 ekrana! Igra teče i dalje, nadate se da ćete stići i do sledećeg, kad čujete zvono na vratima (na teleton se, naravno, na biete obazirali) Šta da sa radi? Pominjući eva lepe reči kojih se setite, pritiskate BREAK i odlezite do vrata da biste ustanovili da nako žali de pita gde je stan porodice Vaš najbolji skor avih vremena je, narevno, otišao u nepovrat

Mora li da bude baš tako? Naravno da na more eko posedujete program „Univerzaina pauza" Učitajte ga pra učitavanje bilo kog (ozbiljno mlaimo bilo kog) drugog programa I, kada želite pauzu pritisnite taster @ (biznis e lli, kako ga naši programeri nazivaju „majmunaki znak ') Ekren će sa potpuno zamrznuti — jedino će sa (eventualno) čuti kontinuiranl ton Kada poželite da nastavite pritisnite bilo koji taeter (na primar onaj koji treba da označi vašu sledeću ekciju) i produžite sa igrom

Kako „Univerzalna pauza" može da bude univerzalna? Principi ovog programa su prilično jednostevni, leko će mnogi pomisiiti da oni odlaze predaleko u „mračna voda operativnog aistema Vaš računar radi u režimu stalnih prekida (interapt) — evaki put kade, na primar, pritisnate naki taster, mikroprocesor dobije maskirani interapt da bi ga registrovao Poseban vid prekida eu takozvani događaji (events) koji eu više informativna priroda i koja računar u normalnoj situaciji ignoriše Pomoću *FX 14 (iii kako je uradeno u ovom progremu, odgovarajućeg OSBYTE poziva) naki od događaja mogu da budu omogućani U tom slučaju mikroprocesor koristi EVNT vektor de odredi adresu na koju će „skočiti" kade događaj nastupi U našem slučaju omogućen je događaj 2 koji: kako piše u uputstvu, nastupa kada je pritisnut naki taster U tom slučaju izvršava se PROG, u ekumuleteru je broj 2 (broj događaja), a u Y registru ASCII kod pritisnutog tastere U okviru tretirenja događaja potrebno je, najpre, sačuvati eva procesorske registre na steku Posle toga nastupa pauza koja je neophodna da bi korisnik otpustio taster @. Zatim računar skanire tastaturu bez korišćenja rutina operativnog sistema — za vrama „događaje" interapt je onemogućen pa ove rutine na funkcionišu Ovo skaniranje je omogućeno testiranjem područja SHEILA koje je veoma nekompletno obrađeno u okviru uputstva za upotrebu Ukoliko je detektovan pritisak na bilo koji taster, računar sa eteka skida potrebna registra i, pomoću RTS vrače kontrolu programu koji ee izvršavao pre pritiska na @

```
 10 REM
 20 REM
 30 REM          PAUSE source
 40 REM
 50 REM     Dejan Ristanovic 1984
 60 REM
 70 REM
 80 REM
100 OSBYTE=&FFF4:FOR I=0 TO 3 STEP 3
200 P%=&900    REM !!! neka druga adresa
300 [ OPT I
400 .INIT
401 \
402 \  Promena sadrzaja EVNT vektora
403 \
404 \    - kada god bude bio pritisnut neki taster racunar ce skociti na PROG
405 \
406 \
500 LDA #(PROG MOD 256)
600 STA &220
700 LDA #(PROG DIV 256)
800 STA &221
801 \
802 \  Dozvoljava se interapt koji nastupa Po pritisku na neki taster
803 \
804 \
900 LDA #14
1000 LDX #2
1100 JSR OSBYTE
1200 RTS
1300 .PROG
1400 PHP
1500 PHA
1501 \
1502 \  Da li je interapt nastupio zbog Pritiska na taster !!! drugog razloga?
1503 \
1504 \
1600 CMP #2
1700 BNE NNOTKEY
1800 TXA
1900 PHA
2000 TYA
2100 PHA
2101 \
2102 \  Da li je pritisnut bas taster @?
2103 \
2104 \
2200 CMP #&40
2300 BNE ANYKEY
2301 \
2302 \  Mrtva Petlja koja korisniku daje vremena da otpusti taster @
2303 \
2304 \
2400 LDA #10
2500 .DUMMY
2600 LDX #255
2700 .DUMMY1
2800 LDY #255
2900 .DUMMY2
3000 DEY
3100 BNE DUMMY2
3200 DEX
3300 BNE DUMMY1
3400 SBC #1
3500 BNE DUMMY
3501 \
3502 \  Rutina koja skanira tastaturu i detektuje Pritisak na taster)
3503 \
3504 \    standardne rutine operativnog sistema ne mogu da se koriste Jer Je
3505 \    za vreme interapta onemogucen drugi interapt (NEI)
3506 \
3507 \
3508 .SCAN
3700 LDX #15
3800 STX &FE41
3900 LDX #1
4000 STX &FF4D
4100 LDX #9
4200 LDA #1
4300 BIT &FE4D
4400 BEQ SCAN
4500 .ANYKEY
4600 PLA
4700 TAY
4800 PLA
4900 TAX
5000 .NNOTKEY
5100 PLA
5200 PLA
5300 RTS
5400 ]
5500 NEXT I
5600 SAVE "Pause" 900 D50 900
```

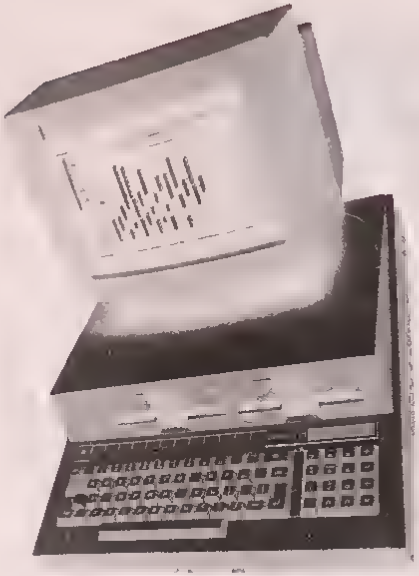
Program možete de otkucate kao i svaki bajzik Posle inicijalizacije programi je neprekidno u računaru sa kojim možete da radite sasvim normalno program će biti obrisan jedino pritiskom na CTRL BREAK dok je posle BREAK potrebna nova inicijalizacije koja se postiže se CALL & BOO

Dejan Ristanović

olivetti M20

OLIVETTI M20

Fabrika Olivetti ima dugu tradiciju u proizvodnji ličnih profesionalnih računara. Tako je još pre dvadeseak godina bila prva sa programom P101 računarom koji je imao 4 K memorije zasnovane na inkolji opruge. Nastavljajući tradiciju u svojoj filijali u Cupertinu u Kalifornij Olivetti je 1979 godine otpočeo razvoj novog ličnog profesionalnog računara. Rezultat M20 zvanično je predstavljen u proleće 1982 godine. Proizvodi se u Italiji a u nas ga zastupa RO DINARA iz Beograda



HARDWARE

M20 je zasnovan na Zilog-ovom potpuno 16-bitnom mikroprocesoru Z 8001 koji radi na frekvenciji 4 MHz. Za njega je napisan poseban operativni sistem PCOS (Profesional Computer Operating System).

Mikroprocesori iz serije Z 8000 omogućavaju segmentirano adresiranje do 128 segmenata po 64 KB (odnosno 8 MB) virtuelnu memoriju 32-bitnu aritmetiku, dinamičku dodelu memorije multiprogramiranje, multiprocesorski rad i još dosta drugih svojstava do nedavno korišćenih samo kod većih sistema. Zajedno sa M20 paralelno je razvijena cela serija od M10 do M60 zasnovana na novoj tehnologiji.

Taj razvoj se i dalje nastavlja i gotovo svakodnevno svojstva i mogućnosti sistema rastu. Osnovni memorijski kapacitet je 128 KB RAM-a a moguća su proširenja sa priraštajima po 32 KB ili 128 KB. Ukupno za sada M20 može imati do 0·5 MB odnosno korisničkih 421 KB neto. Diskretne jedinice od 5¼ inča mogu da budu od 160, 320 ili 640 KB neformalizovano.

Winchester disk sa 11·7 MB neformalizovano moguća je priključiti kao dodatnu periferijsku jedinicu ili ugraditi umesto jedne diskete. Ekran od 12 inča može da bude crno beli sa standardnim ili zelenim fosforom ili u boji sa 4 ili 8 boja. Grafika je sa 512×256 tačkica i 16 prozora. Struktura broja linija i znakova u redu je softwareski promenljiva i može da bude 16×64 ili 25×80. Tastatura ima poseban numerički deo a funkcijski tasteri zamenjeni su kombinovanjem tastera pa je moguće generisati do 256 znakova ili komandi. Poseban naš znaci i tastatura u skladu sa JUS-om.

Osnovna verzija M20 uključuje 1 paralelni priključak za štampač i jedan serijski RS 232. Kao dodaci mogu da budu ugrađeni IEEE488 sa svojih najviše 14 periferika kao i još najviše 4 RS 232 priključka, odnosno 4 strujne petlje. LAN (Local Area Network) priključak omogućava grupno korišćenje kapaciteta na primer datoteka na disku, za najviše 64 računara M20 povezanih u mrežu.

Od velikog broja raznih periferijskih uređaja štampača, plotera traka, kaseta, bušača čitača, itd., koja je moguće priključiti na M20 posebno treba izdvojiti matrični štampač PR 15 sa 120 zn/sek širom od 8 inča 80 do 132 znaka u redu sa gustinom 10, 12 i 16 zn/inču ploterskim mogućnostima oličkom povećana verovatnoća i paralelnim Centronix ili serijskim RS 232 priključenjem.

Za povezivanja sa drugim sistemima na raspolaganju su protokoli asinhroni npr. TTY33 sinhroni BSC npr IBM 2780, bit orjentisani npr. IBM 3275 i X25.

SOFTWARE:

Radi kompatibilnosti sa softwareom drugih proizvođača, M20 koristi proširenje elektronsku ploču sa INTEL 8086, 8 MHz a posluje i verzije se oba procesora na osnovnoj ploči. Tako je omogućeno korišćenje i operativnih sistema MS/DOS (kao IBM PC) CPM/80 (kao i APPLE), CPM/86 i UCSD-p.

Sistemski Software pod PCOS-om obuhvata Assembler Basic (Microsoft interpreteraki, za oko 50% brži od IBM PC-a) Pascal i Forth. Pod ostalim operativnim sistemima mogu da se koriste mnogi jezici uključujući FORTRAN, kao i BASIC sa prevođenjem.

Proizvođač nudi mnogo gotovih paketa od kojih posebno treba istaći

MULTIPLAN — usavršena verzija VISICALC-A (MICROSOFT) za analizu i modelovanja
OLIWORD — obrada teksta (poslovi verzija prevedena na srpskohrvatski)
OLICHART — grafički prikazi rezultata Multiplan analize
OLISORT — sortiranje
OLITERM — komunikacija asinhrona
OLICOM — komunikacije sinhrona
OLEINTRY — unos podataka i obrada
OLISTAT — statistički paket
OLINUM — numerička analiza
OLI3276 — komunikacija IBM 3275

Aplikacioni software pored korisnika radi i oko 400 specijalizovanih software-skih kuća za koja Olivetti periodično publikuje katalog a u nas je urađen jedan iz interaktivne grafike.

Dokumentacija za M20 u ovom trenutku kod zastupnika obuhvata bibliolaku od 56 knjiga koja raste po stopi od 2—3 mesečno, pa ovaj prikaz ličnog profesionalnog računara M20 treba shvatiti kao presek jednog vrlo dinamičnog procesa.

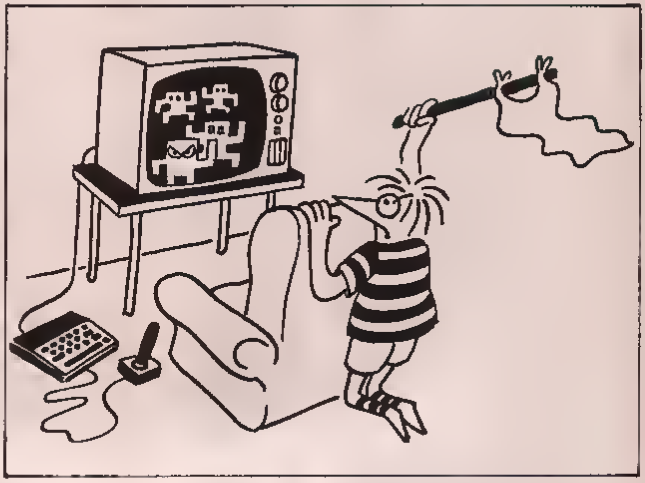
olivetti
Zastupnik DINARA
DINARA-KOMERC
Vlajkovićeva 5 Tel 335-886
BEOGRAD 335-886

„galaksija" bez tajni

Majstorije na računaru

Uputstvo za upotrebu računara „galaksija" koja smo objavili kao podlistak specijalnog izdanja „Računari u vašoj kući" bilo je namenjeno pravoslavno početnicima. Osim njih, „galaksiju" su napravili i mnogi koji su „prevazišli bajzik" i požalili da pišu i mašinska programa. Njima, osim korisna mare sistemskih promenjlivih, uputstvo nija pružalo ništa, nismo mogli da damo čak ni adresa korisnih rutina u ROM-u jednostavno zašto što ona nisu bila tiksirane kada ja uputstvo pisano. Za njih smo ovoga puta pripremili obilja programskih finesa za napradno programiranja računara „galaksija".

Čim je razvoj ROM-a završen ponudili smo svima zainteresovanima njegov disasemblirani listing (i dalje možete da ga dobijate ako na žito-račun „Galaksija" uplatite 200 din i pošaljete nam potvrdu o uplati uz napomenu da želite program GALA005052). On je praktično trenutno postao bestseler ali smo ipak bili svesni da svi oni koji su ga naručili neće moći da izvuku mnogo koristi od njega. Da biste razumeli operativni sistem i bejzik interpretator jednog računara (čak i ako kao u našem slučaju taj program ima samo 4 Kb) morate da poznajete mašinski jezik izuzetno dobro i uložite mnogo rada, strpljenja i što je najvažnije, dosta vremena. Zato smo odlučili da vam u Računarima 2 omogućimo da podete prečicom. Izdvojili smo sistemske adrese koja su nam se činile najkorisnijim, obradili kompletnu aritmetiku pokretnog zareza, proučili za vas korišćenje linkova za naredbe video i (do sada nedokumentovanog) linka u interaptu i sve to sumirali u sledećih nekoliko strana. Obzirom da je sada pisanje mašinskih programa za „galaksiju" postalo moguće na jedan od tri načina (o njima nešto docnije) nadamo se da će podaci koje dajemo biti iskorišćeni na najbolji mogući način za pisanje sistemskih i uslužnih programa i naravno efektnih igara.



Karikature Miloš Arizović

PTR

Dvadeset šesto poglavlje Uputstva za upotrebu računara „galaksija" bilo je posvećeno naredbi PTR. Ona da podsetimo, daje memorijsku adresu neke numeričke ili alfanumeričke promenljive (PRINT PTR X$ na primer daje rezultat 10864 što je isto što i &2A70 prema mapi sistemskih promenljivih). No naredbe PTR može da se koristi i bez adresnog dela: npr. A=PTR ili PRINT PTR (poslednja modifikscija „galaksijinog" ROM-a pred njegovo zaključivanje 3 januara 84.) U tom slučaju naredba PTR daje memorijsku adresu na kojoj je sama smeštena. Šta će nam to? Ako se u memoriji nalazi iole duži bajzik program nije ni malo jednostavno pronaći neko određeno slovo u njemu da bismo ga pomoću BYTE promenili. Ako pre toga slova stavimo jednostavno PRINT PTR i izvršimo program, direktno ćemo dobiti traženu poziciju (na treba zaboraviti da će se ta adresa smanjiti za 9 bajtova ako izbrišemo naredbu PRINT PTR koja računajući i blanko simbol ima isto toliko slova).

Navedimo jedan primar upotrebe naredbe PTR. Dok lista program računar „galaksija" ne očekuje da naide na linijski broj 0. Ukoliko ipak naiđe na njega, računar smatra da je listanje završeno i vraća se u komandni mod prikazavši READY. Kako da kratamo linijski broj 0? Ako želimo da prva naredba programa ima taj broj, ništa lakše: otkucaćemo WORD WORD(&2C36) 0 pa će „galaksija" listati samo prvu liniju programa u kojoj može da piše ime njegovog autora ili možda A=USR(&2C3A) što bi stvorilo utisak da je program pisan na mašinskom jeziku. No šta ako želimo da se listanje programa prekine posle neka druge linije? U prethodnu liniju ćemo na samom njenom kraju dodati naredbu PRINT PTR a zatim je i izvršiti. Dobićemo adresu prvog bajta iza slova R (od PTR). To je bajt &D koji označava kraj programske linije. Iza njega, kao što ćemo videti, stoji linijski broj sledeće naredbe koji možemo da promenimo u 0 primanom naredbe WORD. Ne treba da zaboravimo da obrišemo PRINT PTR naredbu koja je obavila svoju funkciju. Ukoliko bismo ovu naredbu obrisali pre nego što promenimo linijski broj sledeće morali bismo ponovo da preračunavamo adrese jer se sadržaj memorije pomerio.

Naredbu PTR ćete mnogo više koristiti (opisani metod je pritično opasan jer „galaksijin" sistem traženja linijskih brojeva kod GOTO i CALL naredbi očekuje da ovi brojevi rastu) za pisanje programa koji menjaju samog sebe. Jedan od takvih je i „Telefonski imenik" ilustracija Interfejsa 1, koji je objavljen u martovskoj „Galaksiji". Ako sa odlučite na sličnu „avanturu" vodite računa da adresa početka i kraja bejzik programa mora stalno da bude smeštana u sistemske promenljive &2C36 i &2C38. Odstupanje od ovoga može da ima neprijatna posledice po GOTO naredbe i posebno ispravke pojedinih linija.

TAKE

Devetnaesto poglavlje Uputstva govori o naredbi TAKE koja je mnogim vlasnicima drugih računara poznata pod imenom READ. Radi uštede prostora u ROM-u, u bejziku uobičajena naredba RESTORE je zastupljena samo kao poseban slučaj TAKE. Ovoj mogućnosti smo do sada posvetili samo jeden pasus, propuštajući da spomenemo jedan interesantan trik koji je zamišljen još kada je TAKE koncipirena. Ako se iza TAKE nalazi neki broj n (npr. TAKE A,8,100,C), galaksija će potražiti liniju sa tim brojem i postaviti pointer naredbe TAKE na njen početak. Ukoliko se u toj naredbi nalazi (DATA) LISTA, SLEDEĆE TAKE će uzeti podatak iz nje. Ukoliko se u naredbi uopšte ne nalazi lista, računar će potražiti sledeću naredbu, zatim sledeću...

Ovakva struktura naredbe TAKE je prilično fleksibilna i omogućava uštedu RAM-a, sve se dešava u istoj listi. Na prvi pogled, međutim, koncepcija ima veliku manu: simulacija naredbe RESTORE N, gde je N promenljiva a ne konstanta, nije moguća! Zaista, ako otkucamo TAKE A, računar neće pražnjiti listu u kojoj je broj smešten u A, već će promenljivoj A dodeliti vrednost iz prve sledeće liste. Šta da se radi? Mogli bismo narevno da smestimo N u sistemsku promenljivu čija je adresa &2A9D, ali je takvo rešenje hleb sa sedam kora. U ovu sistemsku promenljivu se ne smešta broj linije već adresa njenog početka, posmatrano apsolutno u memoriji. Ovu adresu možemo da odredimo pomoću PTR, ali moramo da je menjamo svaki put kada previmo ispravke u programu, što nije baš mnogo prijatno. Zato je bolje primeniti ovaj trik kada galaksija razmatra naredbu TAKE bejzik interpretator koristi potprogram koji određuje da li je sledeći element numerik. Ako jeste, biće pozvan potprogrem koji izračunava vrednost izraza (ako je taj izraz običan broj, vrednost koja se vraća iz potprograma je sam taj broj, naravno) e zatim se on preračunava u vrednost koju treba smestiti u sistemsku promenljivu. A šta eko umesto TAKE A napišemo TAKE 0+A? Nula plus A je...kao što znamo iz osnovne škole, isto što i A, ali za galaksiju, to ne važi u svakoj konstelaciji. Događaje. Pri analizi naredbe računar nailazi ne broj nula koji prepoznaje kao numerik, podrazumevajući da se radi o simulirnoj naredbi RESTORE. Potprogram koji izračunava vrednost izraza, međutim, nije tako lako prevariti i on će svakako razumeti da je 0+A isto što i A, pa ćemo dobiti naredbu RESTORE A!

```
10 '
20 '              CLEAR
30 '
40 '      DAJE VREDNOST NULA
50 '      NUMERICKIM PROMEN-
60 '          LJIVIMA A-Z
70 '
80 '<
90 ORG &3000
100 DPT 3
110 LINK EQU &2BA9
120 KOMANDA EQU &758
130 PREPOZNAJ EQU &39A
140 DFILE EQU &2B00
150 NUMVAR EQU &2A00
160 LD A,&C3   '  &C3 = JUMP
170 LD (LINK),A
180 LD HL,PROG
190 LD (LINK+1),HL
200 RET
210 PROG
220 EX (SP),HL
230 PUSH DE
240 LD DE,KOMANDA
250 RST &10
260 POP DE
270 JR Z,NAREDBA
280 EX (SP),HL
290 RET
300 NAREDBA
310 LD HL,TABLICA-1
320 JP PREPOZNAJ
330 TABLICA
340 TEXT "CLEAR"
350 BYTE CLEAR>8+&80
360 BYTE CLEAR#&FF
370 BYTE BACK>8+&80
380 BYTE BACK#&FF
390 BACK
400 RET
410 CLEAR
420 POP AF
430 PUSH DE
440 LD DE,NUMVAR
450 CYCLE
460 LD HL,ZERD
470 LD BC,4
480 LDIR
490 LD A,E
500 CP &6B
510 JR NZ,CYCLE
520 POP DE
530 RST &30
540 ZERD
550 WORD &0018
560 WORD &4000
570 >
580 A=USR(&3000)
```

KEY

Odmah iza naredbe TAKE govorili smo o naredbi KEY uz odgovarajuću tebelu u kojoj su se nalazila dva provokativna prazna mesta pa su se mnogi verovatno zapitali šta znači KEY(49) i KEY(51). Pripremajući tabelu...imali smo puno razloga da izostavimo pomenuta dva koda jer u bejziku mogu samo da izazovu zabunu, ali ćemo sada govoriti o njima pošto mogu da budu od

```
&3000: 3E C3 32 A9 2B 21 0C 30
&3008: 22 AA 2B C9 E3 05 11 5B
&3010: 07 07 D1 2B 02 E3 C9 21
&3018: 1C 30 C3 9A 03 43 4C 45
&3020: 41 52 B0 27 B0 26 C9 F1
&3028: D5 11 00 2A 21 3B 30 01
&3030: 04 00 ED B0 76 FE 6B 20
&3038: F3 D1 F7 18 00 00 40
```

velikog značaja pri pisanju mašinskih programa.

KEY(49) testira taster BRK a KEY(51) taster DEL. BRK nema mnogo smisla testirati jer pritisak na njega prekida izvršavanje programa (otkucajte 1D PRINT KEY (49) GOTO 10 i startujte program pokušavajući da pritisnete BRK samo na kratko toliko kratko da ga KEY registruje, ali da se programm ne prekine. Nije lako ali je moguće!

Testiranje DEL je još beznadežniji slučaj: pritisak na njega prekida izvršavanje programa i taj prekid traje sve dok korisnik ne otpusti DEL. U mašinskom programu međutim svakom od ovih tastera možete da dodelite proizvoljnu funkciju ukoliko ne koristite potprogram za testiranje tastature iz ROM-a. U mašinskim programima slavište mogu da se testiraju i prilisci na SHIFT BRK (npr. u „Zamku" i „Jumping Jack-u"), programima koji su emitovani preko Ventilatora ova kombinacija se koristi za prekidanje igre i novi start) i slična nemoguće kombinacije.

Naredbe KEY(0) koristi potprograme za rad sa tastaturom iz ROM a što znači da se jevljaju isti problemi (sasvim očekivani uzgred rečeno) sa testiranjem BRK i DEL ali i jedna nova zegonetka: ako slučano pritisnemo STOP/LIST dok se izvršave KEY (0) počeće listanje programa i samim tim prestati njegovo izvršavanje. Dalje ispitivanje bi nas uverilo da sličen artefekt postoji i kod naredbe INPUT. Nišle ozbiljno ali o tome treba voditi računa u toku pisanja bejzik programa.

INPUT

Kada smo već kod naredbe INPUT, po trebno je da odgovorimo na jedno često postavljeno pitanje: eko koristimo naredbu INPUT X$ tekst koji kucámo ne mora da bude pod navodnicima. Kada su zepravo nevodnici obevezni? Najpre u ∔ tistame (što se vidi i iz spiska naredbi sa primerima iz uputstva za upotrebu) a zatim i u naredbama za dodaljivanje koje bi mogle da izazovu zabunu. Možemo dakle da napišemo X$=GALAKSIJA ali ako u X$ želimo da smeslimo slova Y i $ moraćemo da iskoristimo X$=" Y$" jer X$=Y$ ima drugi smisao! Kod numeričkog INPUT-a imamo puno pravo da otkucamo neki aritmelički izraz umeslo obične konstante. galaksija će ga izračunetli i dodeliti promenljivoj koja je navedena u INPUT-u. U tekstu koji kucamo mogu da se koriste i promentjive i funkcije ali teko će sinteksa bude isprevne u protivnom galaksija će ispisati WHAT? ili HOW? (zavisno od prirode greške) i prestati sa redom. U toku INPUT-a, dakle, biva izvršena i naredba VAL.

VAL

Naredba VAL je neobično moćna i treba je koristiti kad god je to moguće. Potreban nam je, na primer, program koji pretvara heksadekadne brojeve u dekadne. Možemo naravno da izdvajamo slovo po slovo iz promenljive X$, pretvaramo ge u broj i primenjujemo uobičajeni algoritam množenja sa 16, ali postoji i daleko jednostavniji način. otkucačemo

```
1D INPUT X$
20 Y$= & +X$
30 PRINT VAL(PTR Y$)
```

i — to je sve. Ako u odgovoru na INPUT otkucamo 2A0 promenljiva Y$ će dobiti vrednost &2A0 a na ekranu će biti štampan broj 522, prevod 2A0 u dekadni sistem. Ako pišete program koji će vam omogućili da unosile mašinski program dat kao hex dump u memoriju, program popul ovog će vam ušteđeti čitevu trećinu kucanja podataka!

Postoji još mnogo načina da lukavo

rskoristite potprograme operativnog sistema bez poteške da se pozabavite mašinskim jezikom. Za pripremanje pomenutih heks-dampova koji bi bili pogodni za listanje i objavljivanje u „Galaksiji" bilo je neophodno da sastavimo program koji će dati mašinski program pretvoreti u naredbe. Za to je bilo potrebno pretvarati numerike dobijene pomoću BYTE u ASCII vrednosti i od njih kreirati „liste (program koji samoga sebe menja). Za ostvarivanje loga poslužili smo se malim trikom: obrišemo ekran sa HOME, štampamo vrednost koju treba pretvoriti u ASCII karaktere a zatim iz same video memorije (od &2601 pošlo se ne &2800 nalazi znak koji je za vrednosti koje su nam potrebne (do 255) uvek blanko jer su brojevi pozitivni) čitamo jednu po jednu celu u običnim BYTE!

Korišćenje rutine operativnog sistema naravno nosi sa sobom i određene opasnosti da će nas računar „pogrešno shvatiti. Svi oni koji su napisali više mašinskih programa počeli su da „mrise heksadekadno i da koriste heksadekadni sistem za predstavljanje svih programerskih problema. Za njih će dakle biti sasvim prirodno da umesto NEW 1024 otkucaju NEW &400 rezervišući kilobajt za mašinski program koji će biti smešten neposredno iza ispred bejzika. Računar galaksije ih na žalost neće razumeti! Iza NEW i OLD se naime mora nalaziti numerik a ne izraz. NEW &400 će biti shvaćeno kao obično NEW i računar da stvari bude gore neće to prijaviti ni na kakav način. Docnije može do dode do mešanja bejzika i mašinskog programa sa katastrofalnim posledicama po oba. Ako dakle koristimo neku opciju koja nije dokumentovana moračemo najpre da provenmo njeno dejstvo.

Mrtvo polje

Skijaše početnike uče da se zauzlave pre nego što nauče da krenu niz padinu pa ćemo i mi pre nego što počnemo da govorimo o mašinskim programima izložiti koje memorijske zone i sistemske registre treba po svaku cenu čuvati ili bar vratiti u prvobitno stanje.

Postoje dve vrste mašinskih programa jedni se koriste iz bejzika kao potprogrami što znači da posle njihovog izvršavanja „galaksija treba da se vrati u bejzik. Da bi se takva kooperacija ostvarila potrebno je očigledno dobro čuvati sve ono što je interpretatorov radni prostor. Postoje sa druge strane i mašinski programi sasvim nezavisni od bejzika i ROM-a koji na primer predstavljaju interpretatore drugih programskih jezika ili (mnogo verovatnije) kompletne igre. Reklo bi se da ovi programi ne moraju ništa da čuvaju i da mogu proizvoljno da se prostiru u memoriju. No odmah ćemo videti da nije baš tako.

Treba pre svega da razumemo da računar ne izvršava naš program sve vreme. Svake pedesetinke sekunde procesor prekida rad na našem programu i pomaže video-stepenu da generiše sliku na ekranu monitora. Ovu rečenicu pišemo već po ko zna koji put ali za to imamo mnogo razloga iz ličnog iskustva nam je poznato da je vrlo teško naći grešku koja je nastala kao posledica interapta koji programer često

KAKO SE UNOSE MAŠINSKI PROGRAMI

U okviru poglavlja koje je posvećeno melim tajnama računera „galeksije" dajemo doste primere koji treba da ilustruju opisivene tehnike. Neki od tih primere medutim mogu da budu interesantni i kao uslužni programi koje biste redo uneli u računer i koristili u raznim prilikama (ovde najpre mislimo na programe RSAVE i TIME). Za to unošenje će poslužiti hex-dump svakog od tih programa i kratka rutina UTM koju ovde donosimo.

Mašinski program se u računer unosi na jeden od tri načine: pomoću UTM-a ili nekog sličnog programa; pomoću asemblera ili čak ako krajnje soluclja, prako asemblera nekog drugog računere koji je povezan sa „galeksijom. Kade ćemo koristiti koju od ovih solucije? Drugi računar je neobično zgodan eko pišemo komplikovaniji sistemski program (npr. ROM 2) koji će u toku razvoja više puta „krahirati. Posle svakog ovakvog „kraha morećemo da brišemo „galeksijinu memoriju pri čemu izvorni asemblerski program u memoriji drugog računera neće biti izgubljen. Korišćenje asembler programe je zgodno kade pišemo igra ili neke druge uslužne programe i neročito, kada posedujemo ovakav asembler. Program UTM osteje kao rešenje neobično zgodno za unošenje gotovih programe koji su objevljeni u nekom časopisu i čije nas modifikacije ne zanimaju — želimo samo de otkucamo program i de ga koristimo.

Šta je UTM?

UTM je naravno Ultre Tiny Monitor. Kade pogledate koliko je dugaćak verovatno ćete pomisliti de ni ono „Ultre Tiny u nazivu ne ilustruje dovoljno njegove slebosti u odnosu na prave monitor programe. Ali nija baš tako — videćeta da je UTM. za primene za koje je napisan, assvim dovoljan. On omogućave unošenje programa u proizvoljni segmant memorije, testiranje i promenu sadržaja proizvoljne memorijske lokacije i kontrolu sadržeja većih segmente memorije. Evo uputstva za njegovu upotrebu.

Po startovenju programe „galeksija ispisuje njegovo zaglevlje i upitnik. Kada god se pojavi upitnik možete de izaberate jednu od sledećih opcije.

Ukoliko otkucavate neki četvorocifren heksadekadni broj, kome ovog puta ne eme de prethodi oznaka &, računar će prikazati sadržaj memorijske ćelije čije je adresa data tim brojem i proglasiti tu adresu ze tekuću.

Ukoilko otkucavate neki jedno ili dvocifran heksadekadni broj, koma ne eme da prethodi oznaka &, računar će taj broj smestiti u memorijsku ćeliju čije je adresa u tom momentu označena kao tekuća. Zatim će autamatski biti izvršena sledeća opcije.

Ukoilko jednostavno pritienete RET odnosno ENTER tekuća vrednost će biti povećana za jeden, pe će na ekranu biti prikazan sadržaj te memorijeke ćelije koju zatim slobodno možete da menjete primenom opcije 2.

Kako da otkucate neki mašinski program pomoću UTM-a? Startujte UTM sa RUN (ne smete de zaboravite de ukoliko mešinski program počinje od &2C3A, treba de pomerite UTM kucajući NEW NNNN pra kucanje UTM-a ili NEW NNNN OLD NNNN WORD &2C3B, NNNN+&2E3E pre njegovog unošenja sa kasete. NNNN je bar za jeden veći broj od dužine mašinskog progreme, pri čemu treba zaboraviti de NNNN mora de bude dekadni broj). otkucajte početnu adresu programe e zatim bajt po bajt. sedržaje sveke memorijske lokacije. Koristite staino heksadekadni sistem, koji je uobičajen kod programera. Posle unošenje poslednjeg bajte, možete de pritienete BRK i tako prekinete rad ili ponovite početnu adresu i uzastopnim pritiscima tastera ENTER proverite sadržaj memorije koji treba da odgovare hex-dampu. Ukoilko pronadete grešku leko ćete je ispraviti kucanjem novog sadržaja.

```
  5 CALL 248
 10 PRR$(4)
 20 Y$=""
 30 I=0
 40 X$=""
 50 H=1 CALL 160
 60 PRINT ";";
 70 H=BYTE(I):CALL 160
 80 PRINT " ",
 90 INPUT X$
100 IF EQ X$,Y$ I=I+1 GOTO 50
110 IF (BYTE(PTR X$+1)>=0)+(BYTE(PTR X$+2)>=0) CALL 130 BYTE I,H X$='' GOTO 100
120 CALL 130:I=H:GOTO 40
130 X$(0)='$'+X$
140 H=VAL(PTR X$(0))
150 RETURN
160 FOR K=4 TO 1 STEP -1
170 G=H
180 H=INT(H/16)
190 G=G-16*H
200 IF G<10 X$(K)=CHR$(G+48):ELSE X$(K)=CHR$(G+55)
210 NEXT K
220 FOR K=1 TO 4 PRINT X$(K), NEXT K
230 RETURN
240 HOME
250 PRINT AT 480; "GALAKSIJA"
260 PRINT
270 PRINT "UTM  VERSION 1.0"
280 PRINT
290 RETURN
```

potpuno zaboravlja jer se dešava van njegovog uticaja. Da bi računar korektno izvršio interapt, nužno je da ni po koju cenu ne menjamo sadržaj procesorskog IY registra. Ukoliko to učinimo (pa makar se restaurisali samo nekoliko naredbi docnije), rizikujemo da računar posle interapta počne da izvršava program na nekom drugom mestu u RAM-u pa ćemo, pogotovu ako je prostor sistemskih promenljivih poremećen izazvati siguran krah praćen reinicijalizacijom računara i gubitkom programa.

Ma koliko se trudili da naš program bude nazavisan od bejzika, za njegovo testiranje će biti korisno da se ipak oslonimo na njega. Nema toga ko će napisati sistemski program tako da proradi iz prve. Ukoliko se program razvija na galaksiji, posle svake probe ćemo izgubiti program ukoliko ne obezbedimo mogućnost da bejzik interpretator preuzme kontrolu. Čak i ako program razvijemo na nekom drugom računaru, poželećemo da koristimo potprograme iz ROM-a a zahtevaju određene predostrožnosti.

Stek mora da vam bude najveća sveti nja. Ukoliko vaš program ima u nekoj grani više naredbi PUSH nego naredbi POP poviatak u bejzik će biti nemoguć, jer RET nalazi adresu za poviatak na vrhu steka. Isto tako, ako se stek stalno puni računar će se garantovano ugušiti posle izvesnog vremena. Simptomi ove greške su programi koji jedno vreme ispravno rade a onda nastupa krah praćen brisanjem čitava memorije.

Dakle, svaka operacija sa stekom a naročito direktna izmena registra SP treba da bude plod pažljivog planiranja (ovim naravno nikako ne želimo da kažemo da ne treba koristiti stek — bez njega se ne piše ni najkraći programi)koja će imati za cilj da sve bude vraćeno u prvobitno stanje. Vodite na primer računa o tome da poziv potprograma iz koga se ne vraćamo dodaje dva bioja na stek. Kao i svako pravilo i ovo ima izuzetaka: ako se ne vraćate u bejzik pomoću RTN već pomoću JP & 66 ne morate praviše da brinete o steku posle ovog skoka: galaksija ispisuje READY i reinicijalizuje registar SP (u programerskom zaigranu ova adresa se zove tarma pa bi zagrizenji haker pretpodnu rečenicu izgovorili kao ne brini o steku ako računar posle ida na tarmu).

Osim staka adresiranog registrom SP galaksija ima još jedan stek adresiran registrom IX — aritmetički stek čiju ćemo strukturu izložiti kada budemo govorili o aritmetici u pokretnom zarezu. Ovaj stek ukratko sadrži argumente aritmetičkih operacija i ne treba ga pravišle iemetiti. Ponekad će nam naravno zatrebati registar IX (pošto IY i tako ne smemo da koristimo) pa ćemo ga najpre sačuvati (sa PUSH IX) onda koristiti i najzad vratiti u početno stanje (POP IX).

Sistemske promenljive

Što se područja sistemskih promenljivih tiče ne treba menjati poziciju kurzora u memoriji tako da se on nađa van područja &2800—&2A00. poziciju kraja memorije (osim ako ne ugiadujemo proširenje memorije) bejzik pointer poziciju tekuće linija linkova za naredbe i video (osim na

```
&3000: 3E C3 32 A9 2B 21 0C 30
&3008: 22 AA 2B C9 E3 D5 11 77
&3010: 07 D7 D1 2B 02 E3 C9 21
&3018: 1C 30 C3 9A 03 44 49 53
&3020: 50 4C 41 59 B0 29 60 2B
&3028: C9 F1 CD 6A 0A D5 11 00
&3030: 28 19 D1 7E 6F 26 00 C3
&3038: BC 0A
```

```
10 '
20 '          DISPLAY (PPP)
30 '
40 '  DAJE ASCII KOD KARAKTERA
50 '  U VIDEO MEMORIJI CIJA JE
60 '  ADREBA  <PPP> PREMA MAPI
70 '
80 <
90 ORG &3000
100 OPT 3
110 LINK EQU &2BA9
120 FUNKC EQU &777
130 PREPOZNAJ EQU &39A
140 DFILE EQU &2B00
150 KRAJHL EQU &ABC
160 IZRAZ EQU &A6A
170 LD A,&C3     '   &C3 = JUMP
180 LD (LINK),A
190 LD HL,PROG
200 LD (LINK+1),HL
210 RET
220 PROG
230 EX (SP),HL
240 PUSH DE
250 LD DE,FUNKC
260 RST &10 ' POREDI HL I DE
270 POP DE
280 JR Z,NAREDBA
290 EX (SP),HL
300 RET
310 NAREDBA
320 LD HL,TABLICA-1
330 JP PREPOZNAJ
340 TABLICA
350 TEXT "DISPLAY"
360 BYTE DISPLAY##&FF00>B+&80
370 BYTE DISPLAY##&FF
380 BYTE NEGDE##&FF00>B+&80
390 BYTE NEGDE##&FF
400 NEGDE
410 RET
420 DISPLAY
430 POP AF
440 CALL IZRAZ
450 PUSH DE
460 LD DE,DFILE
470 ADD HL,DE
480 POP DE
490 LD A,(HL)
500 LD L,A
510 LD H,0
520 JP KRAJHL
530 >
```

sasvim specijalan način o kome ćemo govoriti nešto docnije) i pomiere početka i kraja bejzika. Promena drugih sistemskih promenljivih na može da ima katastrolalne posledice, ali može da izazove male neprijatnosti (ako na primer promenite sadržaj sistemske promenljive &2BA8, slika će se pomaniti što ćete lako ispraviti).

Ne bismo želeli da pomislite da je najbolje na dnati sistemske promenljive pisanje mašinskih programa pogotovu ako se radi o rutinama sistemske prirode ne može ni da se zamisli bez intervencija u ovom bloku memorije. Hoćemo samo da vas upozorimo da svaka promena neke od sistemskih promenljivih mora da bude rezultat razumevanja njene funkcije i situacija kada se ta funkcija ostvaruje.

Pažljiv pogled na tabelu sistemskih promenljivih otkriva da u njemu ima neiskorišćenog prostora — bajtovi &2A97 &2A98, &2AAA i &2A90 nisu korišćeni, pa bi mogli lepo da se iskoriste za smeštaj nekih podataka koji su našem programu potrebni. Da li je pametno koristiti ih? Nismo sigurni u to — računar galaksija će biti i dalje proširivan pa je sasvim moguće da će naka od varijanti ROM-a ili iskoristiti prazan sistemski prostor. U tom slučaju moraćete da preiadujete vaše programe kako bi ostali kompatibilni i sa proširenjima računara. Teško da će vaš program biti baš toliko dugačak da mu zalale tačno četiri bajta pa vam savetujemo da prostor za podatke potražite na nekom drugom mestu. A jedno od takvih mesta je već poznato — bafer koji se prostire od &2BB6 do &2036. Ovih 126 bajtova su prostor koji se u toku izvišavanja programe svakako ne koristr pa može zgodno da posluži kao zona u koju bi se privremano smeštali podaci. Zašto kažemo privremeno? Ukoliko se izvišavanje programa prekine svakako će biti potrebno da korisnik otkuca nešto da bi program nastavio da radi. To nešto biva smešteno u bafer čime jedan deo njegovog sadržaja nepovratno strada. Treba takođe biti vrlo oprezan sa upotrebom betela ako se koristi naredba INPUT za primanje podataka.

Mašinske programe i podatke koji su za njih potrebni možeta slobodno da smeštate i na kraj RAM-a ali je korisno pre toga promeniti sadržaj sistemske promenljive &2A6A. Područje iznad RAMTOP-a je bezbedno i na njega ne utiču ni naredba za editovanje programa ni operacije sa numeričkim i allanumeričkim nizom. Potrebno je ipak da obratimo pažnju na probleme koji mogu da nastupe ako jednoga dana proširimo memoriju galaksije ili ako program damo nekom prijateilu koji je to već učinio.

Nove naredbe

ROM računara galaksija je koncipiran tako da omogućava relativno jednostavno dodavanje novih naredbi i funkcija. Da bismo dodali novu naredbu ili funkciju nije jasno neophodno da razumemo čitavu strukturu bejzik interpretatora (dovoljan je recept koji ćemo korak po korak dati u okviru ovoga poglavlja) ali nije loše da uočimo nekoliko osnovnih stvari.

Pri analizi bejzik programa računar galaksija u registru DE čuva adresu bajta koji se trenutno obiađuje. Kada pišemo običan mašinski potprogram koji se poziva sa USR registar DE će biti automatski sačuvan na steku. Ako pak želimo da dodamo novu naredbu potrebno je da se uklopimo u sam interpretator tj. da po svaku cenir čuvamo sadržaj DE registra. Ukoliko naša naredba ima i neki adresni deo, potprogram koji je podržava mora da primi i taj deo vodeći računa da registar DE po poviatku u bajzik pokazuje na prvi neobiadan bajt iza.

Kao i DE registar IX permanentno pokazuje stanje aritmetičkih akumulatora i treba ga ukoliko nije vršeno računanje koje viaća rezultat bajziku ostaviti nepromenjen. Naš program mora da sa poštuje i za sve moguće sintaksna greške. Ako pogledate listing ROM-a videćete da je dosta prostora posvećeno proveri sintakse i rutinama za tretiranje grešaka. Tako jednostavno mora da bude. Ukoliko interpretator

VAŽNE ADRESE „GALAKSIJINOG" OPERATIVNOG SISTEMA

Funkcija	Poziv. CALL &
a) Aritmetika pokretnog zareza	
Izraz adresiran sa DE → (IX)	AB2
HL → (IX)	ABC
(HL) → (IX)	A45
(IX) → HL	A6D
(IX) → (HL)	73B
Izraz u zagradi adresiran sa DE → (IX)	781
Sabiranje	B32
Oduzimanje	B1E
Množenja	AE6
Deljenje	AF7
Poređenje elemenata na vrhu aritm. steka	B10
RND → (IX)	C8F
b) Adrese bejzik interpretatora	
Izraz adresiran sa DE (celobrojno) → HL	008 RST &8
Izraz u zagradi adresiran sa DE → HL	A6A
Preskakanje blankova adresiranih sa DE	104
Ako DE adresira zarez, kao RST &8 inače WHAT?	5
KEY(O) uz podršku BRK DEL i LIST	CF5
Karakter iz A na ekran uz &D, &C i &1D	20 RST & 320
Štampanje HL na ekranu kao ASCII niz	6FB
Štampanje alfanumerika adresiranog sa DE terminator je 00 (nema novog reda) ili &D (izdaje se novi red)	937
Lociraj varijablu čije ime pokazuje DE U HL se smešta PTR, ako je C setovan nije varijabla, ako je Z setovan numerička varijabla a u protivnom alfanumerička	125
Prepoznavanje sledeće naredbe	30 RST &30
Stanje aritmetičkog steka	8F6
Slobodna memorija (od ARR$ do bejzika) → HL	183
Snimanje bajta na kasetu Bajt je u A	E68
Bajt sa kasete → C	EDD

(a dodajući naredbe i vi pišete taj interpretator) ne reaguje na grešku, može lako da se desi da računar krahira i da se program izgubi. Kada god u vašoj naredbi sintaksa koju je korisnik otkucao nije ispravna (treba da predvidite naredbu JP &78F (poruka WHAT?) ako je sintaksa ispravna ali argumenti ne odgovaraju — naredbu JP &65A (poruka HOW?) a ukoliko je memorija premala za ono što je korisnik želeo da uradi — JP &153 (SORRY). Pri tome ne morate mnogo da razmišljate o steku (SP će biti reinicijalizovan pre nego što galaksija napiše READY) ali morate da obratite pažnju na registar DE i sistemsku promenljivu &2A9F. Ukoliko se u &2A9F (i &2AA0 naravno) nalazi nula računar će jednostavno izdati poruku WHAT?. HOW? ili SORRY. Ukoliko je u &2A9F nešto drugo računar smatra da mora da prikaže i liniju u kojoj se nalazi greška. Trebalo bi da DE pokazuje na prvi bajt koji nije bio razumljiv kako bi na tom mestu bio štampan upitnik. Ukoliko DE pokazuje neki bajt ove naredbe sve će biti u redu i upitnik štampan na pravom mestu. Ukoliko DE ne pokazuje ispravno brže štampana linija označena promenljivom &2A9F sa upitnikom na kraju. To se na primer dešava ako TAKE listi ne odgovaraju podaci u # listama.

Već smo u prethodnom pasusu videli koliko za nas mogu da učine potprogrami koji već postoje u ROM-u. Moramo da naučimo da se njima služimo što više jer će nam tako nove naredbe biti mnogo kraće pisaćemo ih nauporedivo lakše i brže i jednostavnije ćemo ispravljati greške koje smo napravili. Čitanje „recepta" koji sledi će vas nadamo se konačno uveriti u korisnost od sistemskih potprograma.

Link za naredbe zauzima tri bajta čije su adrese &2BA9 &2BAA i &2BAB. U prvoj od ovih memorijskih ćelija se normalno nalazi C9 što je kôd mašinske naredbe RET dok druge dva nisu važna. Ukoliko želimo da dodajemo nove naredbe treba da promenimo bajt čija je adresa &2BA9 u C3 (kod naredbe JP) a u sledeća dva da smestimo adresu na kojoj se nalazi naš program koji prepoznaje nove naredbe. Ukoliko taj program počinje od &3000 upotrebićemo naredbe

```
LD A &C3
LD (&2BA9) A
LD HL &3000
LD (&2BAA) HL
```

Program koji počinje od &3000 treba najpre da prepozna da li je vreme da se naide na naredbu ili na funkciju. U čemu je razlika? Jednostavno rečeno funkcije se razlikuju od naredbi po tome što se nalaze desno od znaka jednakosti. Tako je na primer PRINT naredba a INT funkcija. Postoje neke reči koje mogu da budu i naredba i funkcija (BYTE, WORD, DOT) ali to ne treba da vas zbunjuje pogled na listing ROM-a će vas uveriti da je svaka od ovih reči u memoriji smeštena po dva puta i da se potpuno različito tretira u zavisnosti od toga da li se trenutno traži naredba ili funkcija. Mi ćemo najpre pokušati da uvedemo jednu naredbu.

Tabela naredbi

Kada očekuje da prepozna naredbu galaksija će na link doći pomoću naredbe CALL i to sa adrese &75B. Obzirom na to na vrhu steka će se u tom slučaju nalaziti bajtovi &07 i &5B. Pošto nam je i tako potrebno da sačuvamo HL registar prva naredba našeg programa će biti EX (SP) HL čime smo u HL registar doveli adresu sa koje je dođeno na link. Sada treba da proverimo da li se u HL registru nalazi broj &75B. To ćemo uraditi tako što ćemo najpre sačuvati registar DE (PUSH DE) zatim u njega uneti ovu konstantu (LD DE &75B) a onda izvršiti RST &10. Time je pozvan sistemski potprogram koji upoređuje registre HL i DE upotrebom svega jednog bajta. Po povratku sa ovoga potprograma Z i C tlagovi su setovani u zavisnosti od rezultata poređenja. Ukoliko je dakle Z tlag resetovan registri HL i DE nisu bili jednaki pa treba vratiti kontrolu bejzik interpretatoru. To radimo tako što najpre restauriramo DE i HL i izvršimo jednostavno RET.

Ukoliko je Z setovan treba da pokušamo da prepoznamo naredbu. Najpre ćemo za svaki slučaj vratiti DE u normalno stanje (POP DE) a zatim ćemo u HL smestiti adresu tablice umanjenu za jedan i preći na sistemski program čija je ulazna adresa &39A.

```
LD HL TABLICA-1
JP &39A
```

Potrebno je još da formiramo tablicu koja se sastoji od naziva bejzik naredbi koje dodajemo i prema čemu se iza svake od njih nalazi ulazna adresa mašinskog programa koji je ostvaruje. Teksti sa, jednostavno, smešla u memoriju slovo po slova ali je ulazna adresa prilično problem. Ona se odnosi na neki program koji se nalazi bilo gde u prvih 16 Kb memorije pa su za njeno smeštanje potrebna dva bajta. Prvi od njih daje značajniji bajt adrese (veoma je važno da uočite da ovo odstupa od standardne Zilogove mnemonike kod koje najpre dolazi manje značajan bajt adresa) pri čemu mu je najznačajniji bit setovan. Ovo setovanje bita je neophodno da bi računar razlikovao kraj imena od početka ulazne adrese.

Posle drugog, manje značajnog bajta adrese sledi ime sledeća naredba koju dodajemo zatim sledeće i tako dalje. Posle poslednje naredbe stavljamo još jednu ulaznu adresu kojoj ne prethodi ni jedna reč. Ta adresa pokazuje brio koji bajt prvih 16 K memorije na koma se nalazi naredba JP &75B. Potrebno je dakle da računar nekako prepozna kraj tabele kako bi znao da nema više novih naredbi.

Naredbu koju dodajemo treba da započnemo jednim POP AF koje će rešiti problem steka koji je poremećen CALL-om na link koji nije završen sa RET. Ovaj POP treba da zaboravimo čim smo ga stavili u program ne treba ga kompenzovati nikakvim dalim PUSH niti smo njim u AF registre doveli neke podatke od značaja.

Sledi instrukcije koje izvršavaju naredbu i njen završetak sa RST &30. Posle ovoga bejzik interpretator će pokušati da prepozna sledeću naredbu koja je adresirana DE registrom.

Koje korisne sistemske programe još možemo da upotrebimo? Potrebno je pre svega da prenesemo neke argumente nove naredbe. Ukoliko je naredbi potrebno samo jedan argument koji sledi iza nje u zagradi i po prirodi predstavlja ceo broj stvar je vrlo jednostavna u okviru tabele ćemo setovati bit 6 prvog bajta ulazne adrese (sada znamo zašto se sve mora nalaziti u prvih 16 Kb — najslarija dva bita ne služe za adresiranje). Izraz u zagradi će biti izračunat računar će naći ceo deo tog izraza i poslati ga našem programu kao sadržaj registra HL. Registar DE će u tom slučaju pokazivati prvi bajt iza zagrade.

Ukoliko iz nekih razloga na žalimo da setujemo bit 6 u tabeli (to će se na primer dogoditi ako realizujemo naredbu koja ima različite moguće tipove adresnog dela) možemo da započnemo naredbu sa CALL &A6A što ima istu funkciju. Ukoliko se argument ne nalazi u zagradi iskoristićemo

RST &8 koji će izračunati vrednost izraza i njen ceo deo smestiti u HL registar. Posle toga naravno može da sledi zarez koji odvaja prvi argument od drugog. Za detektovanje toga služi CALL &5. Ako DE pokazuje zarez on će biti preskočen i sledeći izraz vraćen kao sadržaj registra HL a ako ne pokazuje biće prikazano WHAT? prema pravilima koja smo ranije definisali.

Mozda ćemo poželeti da jednostavno preskočimo blankove i dodamo do sledeće značajne bajtik konstrukcije. To radimo sa CALL &104. Ova rutina će nam koristiti jedino ako želimo samo da detektujemo ono što se nalazi iza naše naredbe u protivnom do sada pomenuti potprogrami obaviće i ovo zaobilaženje blanko simbola.

Pošto dodajemo naredbu nije nam do volje samo da prenesemo argumente — naša nova naredba treba da ima i neko dejstvo. To će retko biti neko jednostavno pomeranje memorije — obično treba kontaktirati sa ekranom ili tastaturom i tu mogu mnogo da pomognu sistemski pozivi.

RST &20 štampa na ekranu karakter čiji je kod smešten u registrar A. Pri tome će se ako je taj bajt &D pre ći u novi red obrisati čitav ekran ako je poslat bajt &C...a obrisati poslednji karakter ako je u A &1D.

Ako se na zadovoljavamo štampanjem jedne jedine cifre poželećemo da koristimo CALL &937. Ovaj poziv na ekranu štampa čitav alfanumerik adresiran DE registrom i to sve do koda &D (izdaja se novi red na ekranu) ili do koda &00 koji ne izaziva novi red. Debagovanju programa će pomoći i mogućnost štampanja vrednosti registra HL kao ASCII niza slova — to se poslize sa CALL &8FB.

Očitavanje tastature

Tastaturu možemo da testiramo na više načina. Filozofiji mašinskog jezika najviše odgovara da se posluzimo memorijskim prostorom &2000 - &2037. Svakom bajtu ovoga prostora dodeljen je po jedan taster prema tablici KEY(N). Najmanja signifikantan bit ovoga bajta je setovan ako je taster pritisnut a resetovan ako nije. Ako na primer, želimo da detektujemo da li je taster A (kod 1) pritisnut, izvršićemo program

```
LD A (&2000+1)
AND 1
JP Z PRITISNUT
```

Mnogo je međutim komfornije koristiti **potprograma za tastaturu CALL &CF5,** na primer, viača u A registar kôd zadnjeg pritisnutog tastera (kao KEY(0)) pri čemu ukoliko je BRK pritisnut računar prikazuje READY a ako je pritisnuto DEL privremeno prekida sa radom. Najneprijatnije je to što STOP/LIST započinje u takvoj prilici najverovatnije neželjeno listanje programa. Sistemski potprogram &CF5 podržava pored ostalog i naredbu REPT za ponavljanje teksta.

Ukoliko želimo samo da računar prekine **sa radom ako je pritisnuto BRK a malo prička ako je DEL u akciji izvršićemo** samo CALL &2FF.

Interakcija bejzika sa mašinskim programima može da se odvija i preko promenljivih A-Z X$ Y$ A(I) i X$(I). Za to je najbolje upotrebiti CALL &125 što locira varijablu na čije ime pokazuje DE. Po završetku rada

```
10 '
20 ' REC (RECIPROCNA VREDNOST)
30 '
40 '    ILUSTRACIJA FLOATING
50 '       POINT ARITMETIKE
60 '
70 <
80 ORG &3000
90 OPT 3
100 LINK EQU &2BA9
110 FUNKC EQU &777
120 PREPOZNAJ
130 NUMIX EQU &A45
140 IZUIX EQU &7B1
150 DELI EQU &AF7
160 LD A,&C3    '   &C3 = JUMP
170 LD (LINK),A
180 LD HL,PROG
190 LD (LINK+1),HL
200 RET
210 PROG
220 EX (SP),HL
230 PUSH DE
240 LD DE,FUNKC
250 RST &10
260 POP DE
270 JR Z,NAREDBA
280 EX (SP),HL
290 RET
300 NAREDBA
310 LD HL,TABLICA-1
320 JP PREPOZNAJ
330 TABLICA
340 TEXT "REC"
350 BYTE REC>8+&80
360 BYTE REC#&FF
370 BYTE NEGDE>8+&80
380 BYTE NEGDE#&FF
390 NEGDE
400 RET
410 REC
420 POP AF
430 LD HL,JEDAN
440 CALL NUMIX
450 CALL IZUIX
460 CALL DELI
470 RET
480 JEDAN
490 BYTE 0   '   BROJ 1
500 BYTE 0
510 BYTE &80
520 BYTE 0
530 >
540 A=USR(&3000)
```

```
&3000: 3E C3 32 A9 2B 21 0C 30
&3008: 22 AA 2B C9 E3 D5 11 77
&3010: 07 D7 D1 2B 02 E3 C9 21
&3018: 1C 30 C3 9A 03 52 45 43
&3020: B0 25 B0 24 C9 F1 21 33
&3028: 30 CD 45 0A CD B1 07 CD
&3030: F7 0A C9 00 00 80 00
```

u HL registru se nalazi apsolutna pozicija promenljive (poput naredbe PTR u bejziku). Carry flag je setovan ako DE na pokazuje na nešto što bi moglo da bude promenljiva. Ukoliko je C flag reselovan Z flag govori o vrsti promenljive setovan Z označava numeričku a resetovan — allamueričku promenljivu.

Sve što smo govorili o dodavanju naredbi ilustruje i program na slici 1. Unesite ga u računar (pomoću hex-loadera) i startujte pa će set naredbi vašeg računara biti dopunjen naredbom CLEAR. Ona dodaje sve numeričke promenljivima (A—Z) vrednost nula, što je neobično korisno pri inicijalizaciji programa (kada normalno otkucale RUN promenljive zadržavaju vrednosti koje su dobile u prethodnim programima ili ako od uključivanja računara nisu korišćena sadrže broj 0.5).

Zašto smo se odlučili za naredbu koja ne opšti sa ekranom na prima ništa sa tastature niti zahteve nake argumente? Jednostavno smo želeli da ilustrujemo sam princip dodavanja naredbi dok za pranošenje argumenata ima vremena — njega ćemo ilustrovati u sledećim redovima.

Nove funkcije

Funkcije se kao što rekosmo uvek nalaze sa desna strane znaka jednakosti kao deo aritmetičkog izraza. Najveći deo funkcija (ovde nije reč samo o računaru _galaksija_) zahteva jedan argument koji se da bi se poštovala logika sintakse bejzika obavezno stavlja u zagradu (kod nekih novijih kompjutera ovaj argument može ali ne mora da se stavi u zagradu ali se njenim korišćenjem eliminiše mogućnost zabune u slučajevima kada je argument čitav izraz). Kada naida na funkciju dakle računar najpre izračunava vrednost izraza u zagradi zatim na tu vrednost primenjuje funkciju i dobijenim rezultatom dalje operiše u aritmetičkom izrazu.

Program za dodavanje funkcija je u mnogo čemu sličan programu za dodavanje naredbi mada postoji i nekoliko razlika. Prva i najvažnija razlika je činjenica da se na ovaj program ne stiže sa lokacije &75B već sa lokacije &777. Na početku programa dakle ne treba u DE registar smestiti &75B već &777 pre nego što se izvrši RST &10 (isto tako na kraju tabele na treba ići na lokaciju RAM-a na kojoj se nalazi JP &75B već nagde treba formirati JP &777 pa se sa kraja tabele uputiti na to mesto.

Možda je najvažnija da se program koji izvršava funkciju završava sa RET ili JP &ABC a ne sa RST &30. Kada koristimo RET a kada JP &ABC? Kada je rezultat neki ceo broj koji se nalazi u HL registru koristićemo JP &ABC da bi taj broj bio smešten na aritmetički stek. Ukoliko smo već računali sa brojevima u pokretnom zarezu tako da se rezultat nalazi na aritmetičkom steku upotrebićemo obično RET. Kako za sada znamo da radimo samo sa celim brojevima naše funkcije će se završavati sa JP &ABC.

Program na slici 2 će nadamo se učiniti da vam dodavanje funkcija buda znatno jasnije. Odlučili smo se za naredbu DISPLAY koja ima jedan argument koji daje poziciju na ekranu. Naredba DISPLAY tada viača ASCII kôd karaktera koji se nalazi na toj poziciji. Ovo jasno uvek možemo da zamenimo sa BYTE (&2800+I) ali nam kao što znamo nije cilj da po svaku cenu dodajemo korisna naredbe koje bi bile složena. Kada razumete princip moći ćete da se okušate i na nekoj drugoj funkciji.

Za sada dakle znamo da dodajemo samo funkcije čiji su argumenti i rezultati celi brojevi. Ovakve funkcije mogu da budu korisne u igrama i sličnim primenama ali su matematički nasilno jeni čitaoci ovih redova svakako poželeli da svoj računar dopuna nekom trigonometrijskom ili eksponencijalnom funkcijom koja im mnogo nedostaje. To nije jednostavno kako se u prvi meh čini ali verujemo da će ovakve funkcije veoma brzo opravdati trud uložen u razumevanje principa njihovog dodavanja. Taj trud će najvećim delom biti utrošen na razumevanje izv obrnute poljske notacije

Obrnuta poljska notacija

Obrnuta poljska notacija, Lukasijevićeva notacija ili jednostavno, RPN, je sistem računanja koji na prvi pogled izgleda sasvim neprirodno (odatle valjda i ono obrnuta u nazivu), ali koji je sa aspekta računara i posebno mašinskih jezika nešto sasvim normalno i skoro bi se moglo reći jedino prihvatljivo. Oni koji su posedovali ili poseduju neki Hewlett-Packardov džepni računar (bio on programabilan ili ne) svakako su upoznati sa principima RPN-a pa mogu slobodno da predu na čitanje sledećeg poglavlja. Preostalima savetujemo da posete nekog prijatelja koji ima ovakav računar i u praksi provere koliko će im ovo poglavlje biti jasno. Proba sa džepnim kalkulatorom je dalako manje naporna od pokušaja da se na mašinskom jeziku (pogotovu ako ne posedujemo asembler) nauči RPN.

Za RPN notaciju je stek neobično važan pojam. Stek možete da zamislite kao orman sa mnogo polica koje se nalaze jedna iznad druge. Na svakoj polici može da se nađe samo jedan predmet (u našem slučaju samo jedan broj). Zato kada god želimo da stavimo broj na donju policu broj koji se u njoj nalazio mora da bude prenesen na prethodnu, broj iz prethodne na policu iznad nje i tako dalje. U teoriji bi stek morao da bude beskonačan što znači da se dodavanjem broja ništa ne gubi iz njega. No računari teško mogu da posluju sa beskonačnom memorijom koja bi morala da bude upotrebljena da se sačuvao beskonačni stek, pa naš orman ima gornju policu. Kada stavimo neki broj na stek broj iz pretposlednje police će preći u poslednju dok će broj iz poslednje police biti za sva vremena izgubljen.

Najniže dve police u steku su posebno privilegovane jer se u njih smeštaju argumenti aritmetičkih operacija. Ako na primer treba da saberemo brojeve 2 i 3 najpre ćemo na stek staviti broj 2 a zatim i broj 3 (time će naravno broj 2 biti pomaknut u sledeću policu) a zatim pozvati program za sabiranje. Da li će on izračunati zbir i staviti ga na stek? Stvari na žalost nije tako jednostavna po sabiranju nam je potreban zbir a ne i argumenti koji bi na ovaj način ostali zauvek na steku. Zato će program za sabiranje najpre spustiti stek za jedno mesto (u gornju policu upisati nulu sadržaj gornje police prenati na policu ispod nje itd. sve dok sadržaj druge police na pomeri u prvu a sadržaj prve na skloni sa steka) sabrati argumente koji su se tamo nalazili na najniže dve police i smestiti zbir na najnižu policu zaboravljajući njen prethodni sadržaj. Ako su se dakle na steku nalazili brojevi 2 3 4 i 7 (tim redom 2 je na najnižoj polici) posle sabiranja će na steku biti brojevi 5 4 7 i 0.

O množenju teško vredi reći išta novo ali oduzimanje i deljenje zahtevaju nekoliko reči objašnjenja. Kod ova dve operacije naime nije svejedno da li će se računati X-Y (X/Y) ili Y-X (Y/X). Zato je u RPN u uvedena konvencija koja kaže da je broj koji se nalazi na najnižoj polici steka uvek umanjilac (delilac) a da je broj iznad njega uvek umanjenik (deljenik). Ako se kao u malopredašnjem primeru na steku nalaze brojevi 2 3 4 i 7 posle izvršavanja oduzimanja će on dobiti oblik 1 4 7 i 0.

```
&2C3A: 3E C3 32 AC 2B 21 46 2C
&2C42: 22 AD 2B C9 F5 21 00 10
&2C4A: 2B 7C FE 00 20 FA F1 C9
```

```
10 '
20 '    USPORENO ISPISIVANJE
30 '
40 '      ILUSTRACIJA  LINKA
50 '           ZA VIDEO
60 '
70 '
80 <
90 ORG &2C3A
100 OPT 3
110 LINK EQU &2BAC
120 LD A,&C3  '  &C3 = JP
130 LD (LINK),A
140 LD HL,PROG
150 LD (LINK+1),HL
160 RET
170 PROG
180 PUSH AF
190 LD HL,&1000
200 DELEY
210 DEC HL
220 LD A,H
230 CP 0
240 JR NZ,DELEY
250 POP AF
260 RET
270 >
280 A=USR(&2C3A)
```

Osim četiri osnovne računske operacije potrebno nam je i poređenje brojeva. Poredi se, naravno, brojevi na najniže dve police steka ali mogu da postoje dva pristupa ovakvoj operaciji. Po jednoj poređenje brojeva ne ismati stek ni u kom smislu (ovako nešto važi kod Hewlett-Packardovih programabilnih kalkulatora kao što je HP41C) a po drugoj poredeni brojevi nestaju sa steka dok se čitav stek spušta za dva mesta uz popunjavanje poslednje dva police nulama (ovakav pristup je zastupljen na računaru galaksija).

Broj u kondenzovanom formatu

Do sada smo smatrali da se na svakoj polici steka u jednom trenutku može naći samo jedan broj što nam je u mnogome olakšalo čitavo rezonovanje. Svaki vlasnik galaksije i mađutim zna da svaki broj u pokretnom zarezu zauzima četiri bajta, a zabuna će biti kompletna kada kažemo da se svaki broj na steku sastoji od pet bajtova!

Da bismo razumeli čitavu zbrku moramo da upoznamo organizaciju galaksijinog pamćenja brojeva. Svaki broj u pokretnom zarezu zauzima četiri bajta (32 bita) memorije nalazeći se u takozvanom kondenzovanom formatu. Kondezovani format međutim, nije pogodan za računanje jer je eksponent broja pomešan sa njegovom mantisom. Zato je pre svake aritmetičke operacije neophodno broj iz kondenzovanog prevesti u normalizovani format. Takav broj će zauzimati 40 bitova od kojih će osam biti neiskorišćeno. Ako po

```
10 '
20 '        ZAMENA  READY
30 '
40 '    ILUSTRACIJA  LINKA ZA
50 '           VIDEO
60 '
70 '
80 <
90 ORG &2C3A
100 OPT 3
110 LINK EQU &2BAC
120 LD A,&C3  '  &C3 = JP
130 LD (LINK),A
140 LD HL,PROG
150 LD (LINK+1),HL
160 RET
170 PROG
180 PUSH AF
190 CP 64
200 JR Z,MIPRO
210 LD A,(RADNI)
220 CP 0
230 JR Z,NOPRINT
240 DEC A
250 LD (RADNI),A
260 BACK
270 POP AF
280 POP HL
290 POP HL
300 RET
310 NOPRINT
320 POP AF
330 RET
340 MIPRO
350 LD A,6
360 LD (RADNI),A
370 PUSH DE
380 LD HL,PORUKA
390 LD DE,(&2A68)
400 LD BC,11
410 LDIR
420 LD (&2A68),DE
430 POP DE
440 JR BACK
450 PORUKA
460 BYTE 64
470 BYTE 39
480 TEXT "GALAKSIJA"
490 RADNI WORD 0000
500 >
510 A=USR(&2C3A)
```

```
&2C3A: 3E C3 32 AC 2B 21 46 2C
&2C42: 22 AD 2B C9 F5 FE 40 28
&2C4A: 11 3A 80 2C FE 00 20 08
&2C52: 3D 32 80 2C F1 E1 E1 C9
&2C5A: F1 C9 3E 06 32 00 2C D5
&2C62: 21 75 2C ED 5B 68 2A 01
&2C6A: 00 00 ED B0 ED 53 68 2A
&2C72: D1 18 E1 40 27 47 41 4C
&2C7A: 41 4B 53 49 4A 41 00 00
```

završetku rada treba zapamtiti neki normalizovani broj pretvorićemo ga ponovo u kondezovani format i smestiti u memoriju. Čemu ova komplikacija kada su svi brojevi mogli da se čuvaju u normalizovanom formatu? Jednostavno zbog uštede skućenog memorijskog prostora.

Za programera bi bilo previše komplikovano da stalno misli na to koliko bajtova zauzima koji broj na steku koliko je ovaj popunjen šta pokazuje registar IX i slično. Zato su obezbeđeni sistemski programi koji omogućavaju da stek zamislimo kao orman sa policama opisan u prethodnom poglavlju. Ukoliko se služite njima, čitaoce vam se

da se na svakoj polici nalazi po jedan broj nećete misliti na to koliko on ima bajtova u kom je formatu i tome slično Zato ćemo stedeće redova posvetiti sistemskim pol-programima za operaciju sa stekom

Najčešće će procesorski registar HL pokazivati na neku memorijsku lokaciju počevši od koje je smeštena neka konstanta koja zauzima četiri bajta CALL &A45 će ovu konstantu pravesti u normalizovani format i smestiti je na aritmetički stek uz njegovo obavezno pomeranje nagore trinveznu funkciju (smeštanje bioja sa dna steka u četiri bajta mamorija na koja pokazuje HL) obavlja CALL &A6D

Kako sa medutim konstante uopšte stavljaju u memoriju? Ako se radi o celobrojnoj konstanti možamo da je smestimo u HL registar i izvišimo CALL &ABC Ovaj poziv će sadržaj registra HL pretvoriti u broj u normalizovanom obliku i smestiti ga na aritmatički stak uz uobičajeno pomeranje inverznu funkciju (smeštanja celog dela broja sa dna aritmetičkog steka u HL uz spuštanje steka) obavlja CALL &A6D

Kako medutim da smestimo konstantu koja nije cela tako da je docnije pranesemo na stek naredbom CALL &A45? Mi nameino nismo trošili prostor na objašnjanja smeštanja brojeva u kondenzovanom formatu (možete li da dedukujete način na koji galaksija smešta brojeve? — nagiadićemo najbolje obiazložene odgovora) jai ovo prevodenje može za nas da obavi operativni sistem primanom jednog tepog tiika Potrabno nam je na primai da ustanovimo kako se u memoriju smešta broj 1 (on je ceo ali na sasvim isti način možamo da postupimo i ako nije) Tada ćemo najpie izvršiti A=1 a zatim FOR t=0 TO 3 PRINT BYTE (PTRA+I) NEXT I Tako ćemo bajt po bajt videti da se broj 1 smešta u mamoriju kao (posle prevodenja u uvek pogodnija heksadekadne brojeva) &0 &0 &80 i &0

Krajnja je vreme da damo i adrese sistemskih rutina koja obavljaju osnovne operacije CALL &B32 sabira CALL &B1E oduzima, CALL &AE6 množi CALL &AF7 dali a CALL &B10 poiedi brojeve (izzultat poredenja biva smeštan u Z i C flagove na uobičajeni način) Da ne ostanemo dužni onima koji žele da pišu programa za igra-sistemski program koji na stek stavlja slučajni broj izmedu 0 i 1 može da se pozove sa CALL &C8F Ukoliko nam je najzad potrebno da izračunamo vrednosi čitavog izraza i smestimo je na stek postavićemo DE registar tako da pokazuje na početak izraza i izvršiti CALL &AB2 Ukoliko je taj izraz u zagiadi (npi ako je argument neke funkcija) upotrabićemo CALL &791

Ukoliko najzad želimo samo da utičemo na znak broja setovaćemo ili resetovati bit 7 u celiji na koju pokazuje IX+1 u zavisnosti od toga da li je broj negativan ili pozitivan

Kao ilustraciju svega ovoga i još jedno podsećanja na mogućnosti dodavanja funkcija dajemo program sa stika 3 On dodaje računaru funkciju REC koju prati argument u zagiadi Funkcija izračunava kao što joj ima i govori iacipročnu vrednost ovog argumanta što i nije nešto naročito korisno ali zato odlično ilustruje rad sa aritmatičkim stekom Namerno se nismo poslužili sistemskim programom &ABC koji je mogao da pranese jedinicu iz HL na stek — smeštanje četvorobajtne konstante u memoriju i njeno pranošenje na stek pomoću CALL &A45 je mnogo instruktivnije

```
10 '
20 '        PERMANENTNI SAT
30 '
40 '     ILUSTRACIJA LINKA U
50 '          INTERAPTU
60 '
70 <
80 ORG &2C3A
90 OPT 3
100 DFILE EQU &2B00
110 INTER EQU &FD
120 Y EQU &2A80
130 HALT
140 LD IY,ENTRY
150 RET
160 ENTRY
170 LD A,(&26B0)
180 CP 0
190 JR NZ,BACK
200 LD A,(Y+2)
210 CP ":"
220 JR NZ,BACK
230 LD A,(Y+5)
240 CP "."
250 BACK
260 JP NZ,INTER
270 LD DE,DFILE+20
280 LD HL,Y
290 DISP
300 LD A,(HL)
 310 AND A
320 JP Z,INTER
330 LD (DE),A
340 INC HL
350 INC DE
360 JR DISP
370 >
380 A=USR(&2C3A)
```

```
&2C3A: 76 FD 21 40 2C C9 3A B0
&2C42: 2B FE 00 20 00 3C 3A 82 2A
&2C4A: FE 3A 20 05 3A 85 2A FE
&2C52: 2E C2 FD 00 11 14 28 21
&2C5A: 80 2A 7E A7 CA FD 00 12
&2C62: 23 13 18 F6
```

Link za video

Pogled na mapu sistemskih promanljivih pokazuje da gataksija osim linka za naredba ima i link za video na adresama &28AC &28AD i &28AE Pre nego što će na ekranu ispisati bilo koje slovo (podrazumeva se da to slovo traba da buda ispisano sistemskim pozivom RST &20 a ne direktno smešteno na ekran pomoću LD ili BYTE) računari izvršava CALL &28AC Ukoliko po pozivu naide na RET računari će se vratiti u ROM i ispisati karakter Ukoliko medutim na ovim adresama nade neki JP naš program će preuzeti kontrolu I karakter će biti ispisan tek poste naše obrade kada računar nada RET na kraju našeg programa u RAM-u

Link za video je u stvari obezbaćen da bi uopšte bilo moguće podići rad sa štampaćem Ukoliko je ovaj priključen trabelo bi sastaviti progism koji kada je neki tlag setovan prenosi svako slovo koje treba da bude ispisano na ekranu na papir posredstvom porta i štampača koji je sa njim povezan Videćemo medutim da se sa ovim linkom mogu postići i neke druge stvari koje će interesovati i one koji ne poseduju štampač Počećamo naravno od najjednostavnija i ne naročito korisne primene koja će nam pomoći da shvatimo drugi primai programa

Da li vam se nekada učinilo da vaša galaksija radi prabrzo? Naravno da nije (to se na dešava čak ni vlasnicima dalako bržih kompjutera) ali ćemo vam prikazati program koji će usporiti ispisivanje poruka Ideja je u tome da nije ni malo teško sastaviti bejzik program koji ća na ekranu usporeno ispisivati neki tekst da bi se na primer isekizovala neka EPP poruka tpak letiki su računari kod kojih se može interveniati tako da sa uspori ispisivanje poruka bejzik interpietatora Pokažita program sa slika 4 nekom vlasniku ZX81 ili Spectruma — računara koji ne omogućavaju nikakav izlazak iz ROM a!

Kako radi ovaj program? Ništa lakše — pia ispisivanja svakog karaktera galaksija izvesno vrama izvršava mrtvu petlju koju možete da skraćujete ili produžavate menjanjem bioja koji se smešta u HL registar na početku njenog radnog dela. Po završetku mrtve petlje jedno RET vraća kontrotu nad računarom bejzik interpretatoru u ROM-u Primetite da je bilo neophodno sačuvati registar AF

Program sa slika 5 predstavlja dalako složeniju primenu linka za video i znatno veća posezanja u rad operativnog sistema Umesto READY računar će ispisivati GALAKSIJA sva dok je ovaj program u memoriji Kako smo to postigli? Pia svakog READY galaksija ispisuje znak koji se sastoji od dva karaktera čiji su kodovi 64 i 39 (označeni u tabeli na strani 18 kao spac). Mašinski program najpre ispituje da li se u registru A nalazi kôd 64 koji označava da upravo traba da buda štampana lava polovina znaka Ukoliko nije stigao kôd 64 izvršava se obično RET i dozvoljava ispisivanje karaktera Ukoliko, medutim, program prepozna karakter čiji je kôd 64 biva proglašeno vanredno stanje u jednu memorijsku lokaciju se upisuje podatak da sledećih 6 bajtova ne traba štampati a zatim sa na sasvim standardan način na ekranu ispisuje tekst GALAKSIJA

Najsloženije je bilo zabraniti računaru da štampa karaktere Da bi ovo bilo moguće proučili smo rutinu za štampanje karaktera u ROM-u i utvrdili da galaksija posle RST &20 poziva dva nivoa potprograma od kojih je drugi link Zato naša rutina skida sa steka prva četiri bajta koja predstavljaju dve adrese za povratak sa potprograma (trebalo bi da svi znamo da adresa za povratak sa potprograma ima 2 bajta a da isto toliko sa steka sklanja jedno POP HL) Računar dakle na završetku našeg programa nailazi na RTN ali sa ono ne viaća na mesto na koma se nalazio CALL koji je pozvao link već u rutinu koja je pozvala RST &20 karakter tako nija ni štampan što smo i želeli

Link za video može da se koristi i za mnogo što-šta drugo (sačekajte samo malo pa ćete naići na program koji se automatski startuje) pa i za jednostavno intervenisanje u video igrama Ima na primer igara kod kojih se igračev broj života ispisuje samo na ekranu pa bi u principu bilo moguće beskonačno produžavati igru zabranjujući brisanja karaktera koji ga predstavljaju Čini nam se ipak da je ovakva primena

sličnih linkova daleko zahvalnija kod drugih računara kod kojih je struktura ekrana komplikovanija (link ze video kod Spectruma bi mogao divno da se koristi da postoji). Autori igara za ‚galaksiju' će svakako, smetnati da je lakše prosto poukovati nešto u ekran nego koristiti sistemski i program koji to radi!

Link u interaptu

Na početku poglavlja koje je posvećeno računaru ‚galaksija' rekli smo da nikako ne treba koristiti registre IY jer je on korišćen kao link u interaptu. Pošlo smo do sada nadamo se stekli sva potrebna znanja o pojmu linka posvetićemo se podrobnije ulozi koja je dodeljena registru IY. Moramo, međutim, da upozorimo da su linkovi koje smo pominjali opasni preme redosledu izlaganja. što znači da je najopasnije baviti se ovim poslednjim pra startovanja svakog programe treba „sors lajl' (izvorni program) snimiti na kasetu. Kada pogrešno završen interapt uguši računar, RESET najverovetnije neće pomegati — biće potrebno de iskljućits i ponovo ga uključite

Pošto je pomogao video stepenu da generiše sliku i povećao brojač sekundi u časovniku (eko je ovaj uključen naravno) mikroprocesor je pri kraju interapta — treba još samo uzeti sistemske registre se steka i izvršiti RETI. Pre nego što to uradi međutim mikroprocesor nailazi na naredbu JP (IY). U registru IY je u toku incijelizacije računare smešten broj &00FD pa mikroprocesor izvršava naredba koje se tamo nalaze. To su naravno četiri POP naredba i RETI

Mi međutim možemo da promenimo sadržaj registra IY tako da on pokazuje neku lokaciju u RAM-u. Tamo treba da se nalazi program koji će se izvršavati svaka pedesetinke sekunde i koji će se završavati sa JP &FD. Pri izradi takvog programe međutim. treba da budemo neobično pažljivi on pre svega mora da se izvršava vrlo kratko jer svaki mešinski ciklus njegovog izvršavanja skraćuje iskorišćenje mikroprocesora za izvršavanje redovnog programe Osim toga za vreme interapta ne sme nikako de se desi da mikroprocesor dobije novi interapt, eko ništa drugo zbog toga što bi se slek tako etalno punio. Da bi to bilo sprečeno na samom početku interapta mikroprocesor automatski izvršava DI. Ukoliko naš program koji se izvršava u interaptu samostalno ili nehotičnim pozivom neke rutine iz ROM-a izvrši EI vrlo je verovatno da će nastupiti veliki problemi. Ako se ovo ne desi ali se interapt ne omogući u razumnom vremenu slika se više neće videti i jedini spas će biti isključenje računara

Osim kratkoće, savetovali bismo vam da u okviru programe koji se izvršava u interaptu ne pozivate potprograme iz ROM-a. jer većina njih u nekim slučajevima može da poremeti normalan tok stvari (odlaskom na rutinu za grešku na primer). Ako. dakle želite da testirate neke tastere u interaptu moraćete da testirate ćelije &2000-&2037 a ne da koristite CALL &CF5

Na slici 6 dajemo primer korišćenja linka u interaptu. Radi se o rutini koje se nekada nalazile u ROM-u ali je morala da bude

```
10 ;
20 ORG &3F00
30 OPT 3
40 LINK EQU &28A9
50 Z693 EQU &7568
60 X343 EQU &39A
70 DFILE EDU &2B00
80 NUHVAR EQU &2A00
90 BYTEA EDU &E46
100 WDADHL EQU &E63
110 ZAAEZ EDU 5
120 LD A,&C3    !  &C3 = JUHP
130 LD (LINK),A
140 LD HL,PAOG
150 LD (LINK+1),HL
160 AET
170 PROG
180 EX (SP),HL
190 PUSH DE
200 LD DE,Z693
210 RST &10
220 POP DE
230 JR Z,NAREDBA
240 EX (SP),HL
250 AET
260 NAAEDBA
270 LD HL,TABLICA-1
280 JP X343
290 TABLICA
300 TEXT "ABAVE'
310 BYTE RSUB%8+&80
320 BYTE ASUB%#FF
330 BYTE BACK>8+&80
340 BYTE BACK%#FF
350 BACK
360 RET
370 ABUG
380 POP AF
390 LD HL,&66
400 PUSH HL
410 RST 8
420 PUSH HL
430 CALL ZAAEZ
440 INC HL
450 PUSH HL
460 CALL ZAAEZ
470 LD B,H
480 LD C,L
490 POP HL
500 POP DE
510 PUSH BC
520 LD B,&40
530 DI
540 LEADER
550 XOR A
560 CALL BYTEA
570 DJNZ LEADER
580 LD A,&A5
590 CALL BYTEA
600 POP BC
610 PUSH HL
620 PUSH BC
630 LD H,D
640 LD L,E
650 ADD HL,BC
660 LD B,&A5
670 CALL WDADHL
680 LD A,B
690 POP BC
700 POP HL
710 PUSH HL
720 ADD HL,BC
730 LD B,A
740 CALL WORDHL
750 POP HL
760 JP &E57
770 >
780 A=USA(&3F00)
```

```
&3F00: 3E C3 32 A9 28 21 0C 3F
&3F08: 22 AA 28 C9 E3 D5 11 56
&3F10: 07 D7 D1 2B 02 E3 C9 21
&3F18: 1C 3F C3 9A 03 52 53 41
&3F20: 56 45 BF 27 BF 26 C9 F1
&3F28: 21 66 00 E5 CF E5 CD 05
&3F30: 00 23 E5 CD 05 00 44 4D
&3F38: E1 D1 C5 06 40 F3 AF CD
&3F40: 68 0E 10 FA 3E A5 CD 68
&3F48: 0E C1 E5 C5 62 6B 09 06
&3F50: A5 CD 63 0E 78 C1 E1 E5
&3F58: 09 47 CD 63 0E E1 C3 57
&3F60: 0E
```

žrtvovana zbog nekih rešenja rada sa alfanumerićima naprekidnom prikazivanju vremene Kada otkucamo i izvršimo ovu rutinu neće se bar u prvi mah. ništa dogoditi Da bismo videli njeno dejstvo otkucaćemo Y$=12 10 15 10"DOT" (zapazite da su minuli i sekundi odvojeni tačkom a ne uobičajenim dveme tačkama) Od toga momenta u desnom gornjem uglu skrana videćemo tekuće vreme Bez obzira da li listamo program ili brišemo ekren vreme će biti prikazivano sve dok na primer ne otkucamo Y$=,12 10 15 10"

Kako je ovaj efekat posignut? Prilično jednostavno. svake pedesetinke sekunda računar prenosi sadržaj promenljive Y$ na ekran ukoliko su ispunjena dva uslova Prvi je da se u Y$ uopšte nalazi vreme (treći simbol treba de budu dve tačka) a drugi da je peti simbol atringa Y$ tačka U ovom programu se javlja i naredba HALT koju obično prenabregavamo u školeme programiranja na mešincu pa smo odlučili da joj ovde posvetimo par reči

Naredba HALT izaziva prekid rada mikroprocesora što obično izaziva pomisao da je ona destruktivna. Mikroprocesor međutim neće zadugo stajati čim nailde prvi interapt njegova aktivnost će se nastaviti To znači da se HALT može prevesti kao ‚sačekaj prvi interapt izvrši ga, pa nastavi sa radom' Kao takva, HALT je neobično korisna naredba ukoliko želimo da radimo nešto se IY registrom ili čak de pranosimo veće blokove ekrena vrlo je korisno upotrebiti naredbu HALT kao neku vrstu sigurnosnog ventila Ukoliko to ne učinimo moguće je de program počne da greši s vremena na vreme u nasko besmislenim trenucima a iskustvo nam govori da se takve greške najteže otkrivaju

Šta bi se desilo de je u momentu kada je mikroprocesor naišao na HALT interapt bio onemogućen (DI)? U tom slučaju bi HALT zaista blokirao računar eli bi pritisak na RESET povratio sve u normalno stanje bez štete po sadržaj RAM-a Dolazak na ‚tarmu' bi, ujedno i omogućio interapt pa bi se pojavile i slika koje do toga momenta nije bilo

Obzirom da govorimo o interaptu neophodno je de sa par reči obradimo NMI — nemeskirani interapt On nastupe kada korisnik pritisna dirku RESET koja se nalazi se zadnje strene ‚galaksije' Nemeskirani interapt se razlikuje od maskirenog po tome što računar kada ga dobije u istom trenutku prestaje sa izvršavanjem naredba koja je trajala i prelazi na program koji se nalazi u ROM-u počevši od &66 Tamo najpre biva reinicijelizovan stek pointer a zatim ispisano READY Posle nemaskiranog interapta međutim ‚galaksija' neće reinicijelizovati sistemska promenljive linkove i sadržaj IY registra. što za prekidnu posledicu ima činjenicu de računar može eli i ne mora da se vrati u radno stanje — ako je na primer, interapt pogrešno organizovan sliks neće biti ni posle njega

Kako se „pakuje" bejzik

Došlo je vreme da se posvetimo perifernjskoj jedinici koju svi vlasnici galaksije redovno koriste — kaselotonu. Pre nego što objasnimo rad sa njom i navedemo adrese dva sistemska potprograme koja će nam omogućiti daleko više od naredbi OLD i SAVE posvetićemo pažnju načinu na koji galaksije pakuje bajzik program u svome RAM-u

Kod galaksije nije zastupljeno tokenizovanje naredbi što ime dobra

i loša strana. Pošto naredba nije pretvorena u poseban kôd (kod Spectruma se, ne primer, PRINT pretvara u &F5) lako je i jednostavno analizirati progam bajt po bajt, ali se zato njegovo izvršavanje unekoliko usporava. Program je dakle, u memoriji smešten onakav kakav jeste, slovo po slovo, uz dva izuzetka.

Linijski broj se kodira kao dva bajta i to u obliku koji proizvodi naredba WORD. Ukoliko je, na primer, prva naredba bejzik programa 1500 GOTO 1550, PRINT WORD (&2C3A) će dati rezultat 1500. Blanko koji računar prikazuje prilikom listanja programe između linijskog broja i prvog slova naredbe nije, u stvari, upisan u RAM-u (tako se štedi memorija), dok se osteli blankovi u okviru linije kodiraju kao i svaki drugi ASCII karakter.

Poslednjeg znaka neka programska linija u memoriji je obavezno smešten bajt &D (13 dekadno), koji predstavlja terminator linije (znak da je ona završena) i pokazuje računaru da su sledeća dva bajta broj sledeće linije. Ukoliko se radi o poslednjoj liniji bejzik programa, iza njenog &D se ne stavlja ništa posebno (kod drugih računara kraj bajzika označavaju jedno do tri nula ili neki drugi specijalni kodovi). Kod galaksija bi ovakav terminetor bio sasvim nepotreban, sistemska promenljiva čija je adresa &2C38 u svakom trenutku pokazuje gde se nalazi prvi slobodan bajt iza bejzik programa. Ukoliko smo ne primer, u memoriju stavili samo bajzik naredbu 10 PRINT A, memorijska ćelija počevši od &2C3A imaju redom sadržaje 0, 10, 80, 82, 73, 78, 84, 32, 65, 13, a promenljiva &2C38 ima vrednost &2C44.

Dobar primer svega ovoga je program Telefonski imenik, koji je objavljen u Galaksiji broj 143. Ovaj program modifikuje samoga sebe dodajući nova naredbe i kodirajući ih na odgovarajući način. On naravno, stalno vodi računa o sistemskoj promenljivoj &2C38, koju povećava posle dodavanja svakog bajta. Pri njegovoj analizi treba jedino imati u vidu da bejzik program ne počinja od &2C3A (tu je smešten mašinski potprogram), već da je sa NEW 60 početak bejzika pomeren za 60 bajtova. Računer to registruje promenom sistemaka promenljive &2C36.

Snimanje na kasetu

Dolazimo, najzad, do snimanja programa i podataka na kasetu. Da bi računar mogao korektno da učita program, moraju da mu budu sveopštene informacije o njegovoj dužini i (kada nije reč o bejzik programu) adresi od koje ga treba upisivati. Zato se ispred svakog programa nalazi takozvani „heder" (zaglavlja) u okviru koga su upisane ove informacija. Pre hedera na traku biva snimljen takozvani „lider" od 96×8 nula bitova koji namaju nikakvu posebnu ulogu — oni jednostavno saopštavaju računaru da sledi program i omogućavaju da se kasetofoni koji imaju automatsku kontrolu nivoa reprodukcija prilagode specijalnim tonovima. Računar pri učitavanju na proverava da li je učitano beš 768 „praznih" bitova pa učitavanja programa na morata da počneta se trekom premotanom tačno na mesto na kome se nalazi početak „lidera" tolerancija jasno, nije velika.

```
&3BAC: 00 00 00 00 00 00 00 00
&3BB4: 00 00 21 FF 29 36 2B 2B
&3BBC: CB 5C 20 F9 21 CD 2B 11
&3BC4: E7 28 01 11 00 ED B8 1B
&3BCC: FE 41 55 54 4F 53 54 41
&3BD4: 52 54 20 50 52 4F 47 52
&3BDC: 41 4D
```

Posle „lidera" nailazi sinhronizacioni bajt &A5 koji se takođe ne unosi u memoriju, on jednostavno govori računaru da je došao kraj „lidera" i da sledi „heder". Heder ima samo četiri bajta: prva dva označavaju adresu od koje program treba da buda smešten u memoriju a druga dva adresu prvog bajta koji će biti slobodan posle unošenja programa. Sledi sem program bajt po bajt. Na kraju snimka nailazi takozvani „ček sum" jedan bajt koji omogućeva računaru da se velikom sigurnošću (ova sigurnost ni kod jednog kompjutera ne može da bude stoprocentna) utvrdi da li je program ispravno upisan u memoriju ili je u toku učitavanja naki bajt ispušten ili detormisan. Šta je dakle „ček sum"? U toku upisa programa u memoriju „galaksija" sabira pročitane bajtove i to po modulu 256 (100+20=120, 120+111=231, 231+30=50=261−256) i toma slično), a „ček sum" ovaj zbir popunjave do 00 (u gornjem primeru „ček sum" bi bio 256−5=251).

Ukoliko bi sa po učitavanju pokazalo da zbir dopunjen „ček sumom" nema vrednost nula bilo bi sasvim sigurno da je program pogrešno upisen, pa bi računer ispisao WHAT? A zatim i READY. U protivnom, vrlo je verovatno (moguća je naravno da je neki bajt povećan za jedan a neki drugi smanjen za 1, što se u konečnom zbiru ne primećuje) da je program ispravno prapisan u memoriju, pa će „galaksija" ispiseti samo READY.

Čak i kada je program pogrešno upisan sa kasete računer „galaksija", zahvaljujući originalnom metodu kodiranja bitova, otaklšava njegovo vraćanja na normalu uz minimalne intervencije.

Na kasetu sa upisuju četvrtasti impulsi koji predstavljaju binarne jedinice i nule. Svaki impuls se sestoji od pozitivne i negativne poluperiode od kojih svaka traje po 0,22 ms. Ukolliko se na kasetu upisuje binarne nula sledi pauza duga 2,46 ms a ukoliko se upisuje jedinica — kratka pauza od 1,1 ms. Iza ovakvog dvostrukog impulsa koji odgovara jednom bitu sliede drugi treći i ostali bitovi bajta koji su kodirani na isti način. Po završetku čitavog bajta sledi dodatna pauza od 5,36 ms koja predstavlja značajno osiguranje, ako se desi da računer zbog oštećenja trake ili nekog drugog problema ispusti neki bit, pauza od 5,36 ms će se „podsetiti" da je bajt završen i on će biti spreman da primi sledeći. Tako će biti ispušteno jedno slovo ali će sva dalja biti ispravna ukoliko nije bilo daljih oštećenja. Računer će jasno po učitavanju ispisati WHAT? pa će obično RUN verovatno naći sintaksnu grašku koju ćemo lako ispraviti. Postoji verovatnoće (mada vrlo mala) da program posle greške u učitavanju bude sintaksno ispravan, all de nekorektno radi. U tom ja slučaju mnogo lakše pokušati novi upis se kasete nego tražiti grešku u imenu promenljiva ili konstanti.

Da bista semi upisalr bajt ne traku traba najpre da mašinskim programom isključite interapt (DI) a zatim da pozovate sistemski potprogram čija je ulazne adresa &E68. Na traku će biti posiat bajt iz registra A.

Učitavanje bajta se kasete se postiže na prilično sličan način — CALL &EDD će nevesti računar da sačeka sledeći bajt (galaksija neće imati ništa protiv ako do naiiaska tog bajta protekne daleko više od 5,36 ms) i da ga prenese u registar C.

Program sa slike 7 je jedan od primera rada se kasetolonom on omogućeva relocireno snimanja programa. Rutina se slika 7 se smešta ne sem kraj memorije kako bi bilo moguće upisivati programe koji se nalaze na njenom početku. Kada izvršite program, vaš računar će posedovati novu naredbu RSAVE koja ima tri argumenta: prva dva odgovaraju argumentima naredbe SAVE koja omogućava snimanja segmenta memorije a treći daje adresu počevši od koje će se na kasetu upiseni program docnije upisivati. Naredba RSAVE &3000, &3500 &500 će snimiti sadržaj memorijskih ćelija &3000-&3500 na traku ali će se ovaj program ubuduće učitavati počevši od &3500 (=&3000+&500) kada upotrebimo obično OLD.

Ovaj program predstavlja dobar uvod za sledeće poglavlje. Želeli smo da navedemo dve primene koje bi trebalo da budu ne samo poučne nego i korisne. Prva od njih će biti automatsko startovanja programa!

Automatsko startovanje programa

Sinclairovi računari omogućavaju da program koji sa učitava se kasete počna da se izvršava bez ikakvih daljih intervencija korisnika. Obrazloženje potreba za ovim automatskim startovanjem je obično tvrdnja de je za početnika teško da pamti naredbu RUN što i nije naki razlog (ako već mora da pamti naredbu LOAD" "RUN valjda ne predstavlja toliki problem). Da su oni koji propovedaju automatsko startovanja iskreni, rekli bi da ono predstavlja dobar način zaštite od neovlašćenog presnimavanja programa. Galaksija nikada nije bila pristalice nekih zaštita programa, najpre zbog želje da softver što više kruži medu njenim čitaocima. Dakle, razlog zbog koga nevodimo informacije vezane za autostart programa nije naše želja da stimulišemo čitaoce da nam šalju zaštićene programe (na kraju poglavlja ćemo reći i kako se ovakve zaštite „probijaju") već verovanja da je autostart lepa „caka" koja ilustruje mnogo stvari o kojima smo do seda govorili.

Ideja je prilično jednostavna: kada računar „galaksija" primanom naredbe OLD učita program on će pokušati da ispiše READY. Ukoliko je u link za video upisan poziv nekog programa, kontrola će biti predate tom programu koji će sa dekle stertovati odmah po učitavanju. Ostalo je još pitanja kako da postignemo da se sadržaj bajtova &2BAC-&BAE promene u toku učitavanje programa. Samo na jedan način — naš program treba da počne da se učitava od adrese &2BAC a njegova ulazne adresa treba da bude ne samom početku. Posle ovakva intervencija nećemo jasno više moći da se oslanjamo na RST &20 sistemsko skrolovanje slika i slične pomoći operativnog sistema. Zauzvrat, čak ni prilisak ne RESET neće prekidati naš program — i posle nemaskiranog interapta „galaksija" će pokušati da ispiše READY i ponovo započeti izvršavanje našeg programa.

Sva ovo ilustruje program sa slika 8 dat kao hex-dump. Unesite ga u memoriju (a zatim (pod uslovom da se u memoriji već nalazi startovan program sa slika 7) otkucajte RSAVE &3BAC, &3BDD, — &1000.

isključite i uključite računar pa otkucajte OLD i startujte kasetu na koju ste maločas snimili program. Po njegovom učitavanju na ekranu će ta videti jednu kratku poruku koju će RESET ponovljati.

Zamislite da vam je neko dao program sličan ovome i to na kaseti. Kako da ga presnimite ako ne želite da sa upustite u metod koji uvek daje dobre rezultate presnimavanja uz pomoć dve kasetofona. Ne brinite nije mnogo teško. Najpre utvrdite njegovu početnu (MMMM) i krajnju (NNNN) adresu. Otkucajte OLD 200 i učitajte program koji će ovim biti relociran za 200 bajtova. Zatim koristeći program sa slike 7 otkucajte RSAVE MMMM+200 NNNN+200 —200 i mirno snimite program na novu kasetu.

Može li da se napravi bolja zaštita? Može ako sastavite program koji će biti toliko dugačak da za njegovo relocirano upisivanje nema mesta (npr. program koji bi sa prostirao od &2BAC do &3FFF) što možete da učinite jedino ako posedujete memorijsko proširenje. Takav program će biti prilično dobro zaštićen od onih koji nemaju toliku memoriju ali oni sa hardverom poput vašeg i dalje neće imati problema. A znači će sa i jedan deo lukavijih sa RAM-om od 6 Kb. sastavićе program za prepisivanja koji će počinjati od &2A00 i smestiti ga u područje numeričkih varijabli koja u toku učitavanja programa nija mnogo bitno.

Ukoliko najzad. sastavljate program uz pomoć nekog drugog računara sa mnogo memorije ili diak jedinicama nema prepreke da napišete rutinu dugu nekih 54 kilobajta koja će samom svojom dužinom biti prilično dobro zaštićena (možete li da sastavite program koji bi presnimio i ovakvo nešto? Ne zaboravite da iza svakog bajta na kaseti može da posloji i pauza duža od 5 36 ms) ali takva rutina neće biti baš mnogo komercijalna jer će njeno učitavanje trajati

Funkcija TIME

U akcionim igrama na kraju se igrači obično rangiraju po osvojenim poenima koji su pored ostalih parametara, funkcija utrošenog vremena. Računar „galaksija" kao što znamo ima časovnik koji je vrlo pogodan za prikazivanje vremena, ali na žalost nije pogodan za merenje broja sekundi koje su protekle između dva događaja. Program sa slike 9 dodaje vašem računaru funkciju TIME koja pravara vrema iz Y$ u sekunda. Ovaj program ilustruje skoro sve o čemu smo govorili u poslednjim poglavljima dodavanje naredbi prijavljivanje grešaka (ako sadržaj promenljive Y$ nije vrema biće prijavljeno HOW?) rad sa brojevima u pokretnom zarezu uključujući i smeštanje konstanti

Svi „galaksijini" bagovi

Redovni čitaoci naših napisa u „Galaksiji" su se u poslednje vreme upoznali sa pojmom bag ali neće biti loše da u cilju dobrog komjuterskog obrazovanja još jednom kažemo nekoliko reči o njemu.

Bag je greška u operativnom sistemu računara koja se ispoljava u nekim specijalnim situacijama kao reakcija na neku na prethodnu akciju korisnika. Ne treba misliti da je bag trenutak kada računar ispiše SYNTAX ERROR ili WHAT? — tada je sve pod potpunom kontrolom operativnog sistema. Posledica baga su obično dalako teže gubitak ilnija programa ili sadržaja čitave memorija pogrešan ispis na ekranu ili gubitak njegovog sadržaja, remećenje bitnih sistemskih promenljivih.

Svi računari imaju bagove koji predstavljaju manju ili veću smetnju njihovim korisnicima. Izuzetno kod nekih računare sa zatvorenim operativnim sistemom (bez PEEK i POKE) bagovi mogu da budu korisni jer otvaraju pristup područjima memorija koja su inače labu za korisnika. Proučavanjem i sistematizacijom bagova računara HP41C članovi američkog PPC kluba su razvili sintatičko programiranje koja im je omogućilo kompletnu kontrolu ovog čuvenog džepnog računara

Što se kućnih kompjutera liče njihovi bagovi prave samo probleme. BBC mikroračunar i dalja ima bagova (najozbiljniji od njih praktično onemogućava korišćenje zaraza u naredbi PRINT) premda njegov operativni sistem predstavlja dvanaestu verziju istog programa. Spectrum (pogotovu stariji modeli) ima pregršt bagova od kojih je na svu sreću malo pogubnih naredbe za editovanje kod TRS 80 izazivaju bazbiojne probleme kod neiskusnih korisnika koji teže da primena sve što su našli u uputstvu za upotrebu. Nija ništa bolje ni kod drugih kompjutera

Iako smo uložili dosta truda da pronađemo i eliminišemo sve „galaksijine" bagove, nismo sa nikada zavaraali nadom da ih neće biti. Teši nas jedino činjenica da su sve „galaksijina" bagova pronašli oni koji su radili na njegovom razvoju — sudeći po vešim pismima nijedan čitalac „Galaksije" nije naišao na neki bag svoga računara. Mi ćemo naravno navesti pronađena bagova da bi se čitaoci ovih redova čuvali situacija koje mogu da izazovu probleme

BUG 1 Otkucajte punu programsku liniju (obaiažja 10 i slova A sve dok računarav bater na buda pun, na primer). Zatim otkucajte EDIT 10 i keda sa linija pojavi na ekranu, pritisnite DEL nakoliko puta a zatim i ENTER Prittsnite LIST — ko je rekao da se u memoriji „galaksija" nalazi neki program?

BUG 2 Otkucajte linju koja će gisaiti 10 A= zatim tridesetak otvorsnih zagrada, pa neki broj, pa isto toliko zatvorenih zagrada a zatim startujte program. Posiedica su ista kao kod baga 1. Napominjemo da će normalan broj od dvadesetak otvorenih zagrada izazvati normalnu reakciju HOW? BUG 3 Ovaj bag je toliko dobro sakrivan da još nismo uspali da ga lociramo U nekim retkim situacijama (uzroci da ponovimo još nisu poznati) naredba za rad as numeričkim nizom A(I) će izazvati poruku SORRY, PRINT MEM daja neki negativan broj. Problam se rešave kucanjem naredba ARR$(−1), posle koja sva postaje normalno

Svi ovi bagovi bi jasno mogli da sa isprave odgovarajućim promenama ROM-a, ali ove promena verovatno neće nikada biti izvršena. Odgovor na pitanja zašto?' je sasvim jednostavan: ako promenimo ROM adrese svih sistemskih potprograma (pa i onih koji su pomanuti u ovom poglavlju) će se promeniti što znači da programi neće biti kompatibilni sa oba verzija ROM-a. Tako ćemo najverovatnija ostali kod 28 verzija ROM-a koju smo proglasili konačnom (PRINT BYTE (55) će vas uveriti da vaš računar ima 28 verzija ROM-a).

I uputstvo za upotrabu „galaksija" nija lmuno na bagove u mapi sistemskih promenljivih na strani 26 postoje dve sistemske promenljive na adresi &2BB1 ali će svakome iola pažljivom čitaocu biti jasno da „registar za REPT", zauzima ćelju &2BB4. Osim toga, horizontalna pozicija teksta" zaista ima inicijalnu vrednost 11, ali dakadno a na heksadakadno. Za autora ovoga teksta je najbolnija da kaže da program sa dna strana 7 ima grešku i to na štamparsku nego logičku, on dobro radi za vrednosti iz primere ali čim bi sa putovanje odužilo preko jednog časa. Obzirom da nam se čini da svaki programer početnik (baš tako) može da napiše ovakav program tako da radi ispravno ovda na dajemo novu verziju

Tima smo došli do kraja ovoga prilično opširnog poglavlja koje je posvećeno računaru „galaksija". Autor ovoga teksta duguje veliku zahvalnost Voji Antoniću konstruktoru „galaksije" za pomoć pri izboru tema koje će biti obrađena, seradnju na pripremi primera upotreba raznih jahnika i posebno, za sve podatke o ROM-u koja niko drugi nija ni mogao da zna. Zahvaljujući ponajviše ovim podacima naša galaksija" je, čini nam se postala jedan od vrlo retkih računare koji kako rekosmo i u naslovu poglavlja „nama tajni"!

Dejan Ristanović

ČETRNAEST GODINA ELEKTRONIKE

Zagrebačka Tvornica računskih strojeva jedan je od jugoslavenskih pionira u primjeni mikroelektroničkih i logičkih sklopova u birotehničkim i AOP uređajima. Već 1966. god. razvijen je na bazi DTL int. sklopova prvi elektronički stolni kalkulator; razvojni modeli koristili su feritnu memoriju, ali u proizvodnoj seriji ugrađivani su već, tada još sasvim novi, MOS shift-registri. Razvoj novih tipova uvijek je pratio najnovije kretanja na svjetskom tržištu komponenti i gotovih proizvoda. Među prvim firmama u Evropi i TRS je počeo ispitivati mogućnosti mikroprocesora prve generacije, kao rezultat takvog rada pojavio se i fakturirni stroj TRS-701 (kasnije 711) sa INTEL-ovim 4-bitnim mikroprocesorom. Na području 8-bitnih procesora u ranoj je fazi sagledana važnost „single-chip" tipova za masovnu primjenu u biro-opremi, pa je jedan niz serijskih proizvoda, gotovo cijeli kalkulatorski program, zasnovan na FAIRCHILD-ovom „F-8" kojeg su za TRS-ove potrebe u velikim količinama izrađivala u posebnim varijantama 3 svjetska proizvođača *(sigurno jedinstven pothvat u Jugoslaviji!)*.

Polovicom sedamdesetih godina nabavljeni su razvojni sistemi za 8800-familiju i se Z80, a rezultat rada a tim- još i danas aktualnim mikroprocesorima, su računski sistemi TRS-702 i 703 kao i periferni uređaji matrični štampači TRS-825/835 i video-terminali TRS-828/838. Danas mogućnosti 8-bitnih procesora još ni izdaleka nisu iscrpljene i mnogi novi TRS-ovi proizvodi koriste hardware i sistemski software za njih razvijen tokom vremena, ali već je započeo i rad na 16-bitnim sistemima, jer su u okviru srednjoročnih razvojnih planova definirani i proizvodi u kojima će procesna „snaga" 16-bitnih mikroprocesora doći do punog izražaja. Paralelno s razvojem elektroničkih sklopova odvija se razvoj programske podrške grupa za podršku u okviru teh sektora broji dvadesetak ljudi i trajno širi područja primjene naše AOP-opreme.

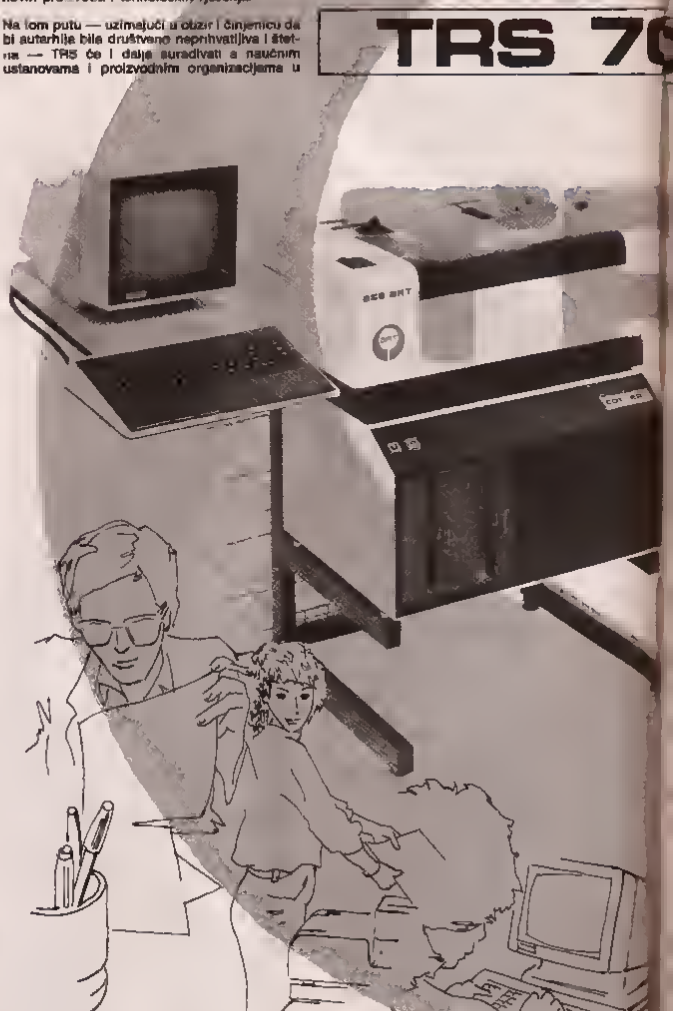
U vezi s novim malim sistemima TRS-703 i 801 donešena je (većim dijelom već i realizirana) odluka da se izrade vlastiti sistemski programi. Operacioni sistemi za „703" i „801" su već gotovi, a sada se radi na PL-1 compileru. Na ovu varijantu smo se odlučili ne samo zbog uštede deviza, koje bi morali trošiti se nabavku u svijetu rasprostranjenih OP-sistema CP/M ili MS-DOS i pripadnih compilera nego i u želji da korisnicima pružimo sistemski software optimiran se naše male sisteme i se profesionalne aplikacije.

U skoroj budućnosti treba pristupiti realizaciji DBM (Data Base Management) software-a, imajući u vidu i neke specifične aplikacija i našu viziju daljnjeg razvoja računarske tehnike u smjeru distribuirane obrade i lokalnim mrežama.

Potpuna modularnost na nivou hardware-a i sistemske i programske podrške omogućava TRS-u da svoju liniju proizvoda prilagodi potrebama naručilaca, a u izgradnji je i takva servisna organizacija koja će na zalsta profesionalan način moći održavati isporučene uređaja u cijeloj zemlji. Poredi dva glavna pravca malih računskih sistema i perifernih uređaja, TRS će i nadalje proizvoditi uredsku opremu (kalkulatora), a a mnogo ambicija ušao je i u područja elektromedicinske opreme (za sada proizvodom „TRS-GLUKOTEST" — uređajem za mjerenje koncentracija šukoze u krvi) S posebnim ponosom u TRS-u se gleda na činjenicu da su svi naši proizvodi rezultat vlastitog razvoja, smišljene i dosljedno provođena kadrovska politika omogućava TRS-u da trajno okupija mlade stručnjake i da ostane jedan od rasadnika stručnih kadrova u zagrebačkoj regiji. Iorili-

ja je našeg vremena da se donedavno vlastiti razvoj manje isplatio od kupnje licenci ili pukog sastavljanja uvezenih podsistema. U TRS-u se nadaju da će sadašnji napori za stabilizaciju naše privrede smiriti one proizvođače koji s vlastitim resursima dolaze do novih proizvoda i tehnoloških rješenja.

Na tom putu — uzimajući u obzir i činjenicu da bi autarhija bila društveno neprihvatljiva i štetna — TRS će i dalje surađivati s naučnim ustanovama i proizvodnim organizacijama u SRH i šire, poduprući napore da se između proizvođača elektroničkih uređaja, a posebno AOP opreme, postignu dogovori o usklađenju proizvodnih programa i da se što više ovih tehnički i tehnološki vrlo složenih uređaja realizira kao zajednički proizvod. U tome i kroz



kooperaciju s odabranim stranim partnerima dugoročno vidimo jedinu šansu za plasman na inozemno tržište i zadovoljenje potreba za devizama. Učešće uvoznog repromaterijala u TRS-ovim proizvodima je oko 10% (po vrijednosti), a budući da se opravdano može govoriti o supstituciji uvoza trebalo bi o toj činjenici voditi računa u gremijima u kojima se raspodjeljuju strana sredstva plaćanja.

Najatraktivniji proizvod Tvornice računskih strojeva svakako ja

3 MALI POSLOVNI SISTEM

Prateća ja cijena obitelji suvremenih malih kompjutera opće namjene. Hardware je modularan i zasnovan na 8-bitnom mikroprocesoru Z80 i snažnim periferniim upravljačkim jedinicama. U osnovnoj konfiguraciji on porad 64 K RAM ima i dvija diskovna jedinice (Floppy-diskete) a kapacitetom od 2.2 MByte Bitna svojstva tog kompjutera određuja njegova sistemska i aplikativna programska podrška za 703" razvijen je i vlastiti operacioni sistem (svakako jedinstveni slučaj u Jugoslaviji!) a dovršava se i prevodilac (complier) za PL-1 viši programski jezik. U asortimanu korisničkih programa najznačajniji su programi za vođenje materijalnog i financijskog knjigovodstva, osobnih dohodaka i saldo-konti, kao i programski paket za organiziranje i praćenje proizvodnje i skladišnog poslovanja. Tu su zatim programi za obradu teksta i za emuliranje raških IBM-ovih alnhronih terminala. U osnovnoj konfiguraciji mali sistem „703" obuhvaća centralnu jedinicu s oba diska, jedan štampač TRS-835 i video-terminal TRS-836. U pripremi je model 901" sa četiri uplana mjesta i „multi-task/multi-user" načinom rada.

TRS 703
TRS 703
TRS 703
TRS 703
TRS 703
TRS 703
TRS 703
TRS 703
TRS 703
TRS 703
TRS 703

TRS 835

SERIJSKI MATRIČNI ŠTAMPAČ

Serijski matrični štampač TRS-835 je izlazna jedinica za ispis podataka. Predviđen je za rad kao štampač — terminal isključivo u prijemnom modu odnosno kao RO štampač (RECEIVE ONLY).

Sve su funkcije štampača mikroprocesorski upravljane i nadzirana radi poetizanja i održavanja zadanih tehničkih karakteristika. Kod projektiranja i proizvodnja serijskog matričnog štampača TRS-835 korištena su suvremena radrijska saznanja i najmodarnije tehnološka dostignuća, što štampaču osigurave vrhunsku kvalitetu, veliku trajnost, sigurnost i pouzdanost u radu

TEHNIČKI PODACI

Brzina ispisa	180 zn/sek
Broj znakova u ratku	132 znaka
Gustina ispiza	2,54 mm/zn
Ispis u oba smjera a optimiranjem puta	
Matnos ispisa	9×9
Veličins znaka	2,54×1,97 mm
Inkrementalni vertikalni pomak formulara	0,529 mm (u obs smjera)
Brzina pomaka formulara	101.6 mm/sek
Gustina ispisa radaka	6 radaka na 25 4 mm ili 6 redake/inchu
Fond znakova	96 zn (JUS III ASCII)
Vrpca velika trajnosti u kaseti	
Veličina prihvatne memorije	274 zn (do 4 K opcion)

Interface prema izvoru podataka.

— CENTRONICS paralelni '
— V 24 (RS 232 c) serijski
— DEC LA 180 paralelni

REFERENCE

Štampači TRS-816' TRS-825 TRS-t250 J i TRS-835 proizvedeni u Tvornici računskih strojeva provjereni su i lapitani s uspejehom u radu sa sistemima alljedećih proizvođača ISKRA DELTA IBM, OLIVETTI

TRS 836

ASINHRONI VIDEO TERMINAL

TRS-836 je suvremeno oblikovani univerzalni video-terminal koji se zahvaljujući primjani modernih mikroprocesorskih komponenti i koncepta po koma se sve funkcije uspostavljaju programski. e na kruto ožičenom logikom može primjeniti u svim aplikacijama u kojime se traži univerzalni asinhroni terminal Pored ostalih interesantnih svojstava posebno treba istaknuti

— mogućnost priključka hard-copy štampača (preko RS 232 C medusklopa)
— mogućnost povećanja video RAM-a do 12 K znakova (6 potpunih stranica) opcija
— mogućnost prikazivanja tzv „business graphics" (na u lokalnom modul)
— mogućnost simuliranja drugih rasprostranjenih asinhronih terminala (IBM 3101 standardno, DEC VT 100 po posebnoj narudžbi)
— mogućnost izravnog povezivenja s host om udaljenim do 1200 m preko RS 422/a medusklopa.

Terminal se sastoji od jedinice s ASCII tastaturom i akransko/logičke jedinice s kadodnom cijevi dijagonale 30 cm (12 inča) sa zelenim tostorom srednje perzistencije. Raspored znakova u tastaturi izmjanjiv je u dogovoru s naručiteljem TRS-836 s različitim sistemima komunicira preko asinhronog medusklopa po RS 422 III RS 232 C za izravnu III udaljenu (remote) vezu odnosno preko 20 mA strujna petlje za izravnu vezu

Ostala svojstva TRS-836:

— radi u znakovnom III blok modu
— brzina prijenosa podataka ja od 110—9600 Bita/sek
— korisnik može odabrati lokalni mod rada bez intervencije sistema lo se može koristiti za učenja operatera, krairanje tormata III editiranje poristaka na akranu, adresiranja cursor-a tunkcija brisanja, samostalni pomak na novi red scroilng-up, prijenos bloka podataka i formatiranja polja
— atributi tormiranih polja mogu biti
  a) zaštićeno / nezaštićeno
  b) vidljivo / nevidljivo
  c) treptajuće / netreptajuće
  d) pojačani intenzitet / normalni intanzitet
  e) modificirano / nemoditicirano
— znakovi su u metrici 7×9, 24 reda a po 80 znakova 25 rad ja za poruke operateru o radu terminala (sistemski red)
— tastatura je argonomski optimalno riješena odvojena od sistema i operater ja može prilsgoditi svom položaju
— akran se može zaokretati oko horizontalna osi i tako postaviti u najugodniji položaj za operatera
— ESCAPE sekvence za izvršavanja specifičnih funkcija koja se na mogu aktivirati preko kontrolnih znakova.

Terminal svojim funkcionalnim karakterisilkama amulira IBM 3101 (model 2×) terminal i može ga u potpunosti zamijeniti.
Terminal je razvijen za rad s IBM S/370 (uključujući 303×), 4300, 6100 i B/1 programskom podrškom

TVORNICA RAČUNSKIH STROJEVA - ZAGREB

41000 ZAGREB — Braće Kavurića 21 — p. p. 02-646 — Brzojav: TRS Zagrab
Telax 21434 — Centrala 447 111, 447 001 447 692 — Prodaja 411 302
Sektor podrška Poslovnica Zagrab, Šaterava 2, 41000 Zegreb tal 212 900, 212 911, — Poslovnica Beograd, 27 marta 35, 11000 Beograd, tal 327-747, — Poslovnica Ljubljana Linharteva 11a, 61000 Ljubljana tal 327-470, — Poslavnica Rijaka, Hotel „Royal" Opatija, Dražice 3, tal 712-544

Komponente za memorijsko proširenje

Konstruktorski hod po mukama

Za konstruktore i hobiste čini nam se nikada nisu bila lepša vremena — čipova je čak i na svetskom tržištu sve manje a cene vrtoglavo rastu. Nesvesni rekordi u skokovima cena drži Hrvatski čip 8255 koji je sa neke tri funte za dva meseca skočio na 19 funti! On ne žalost, ipak ključnu ulogu u našem programatoru EPROM-a a vrlo brzo ćemo mu se kada budemo radili paralelni interfejs za štampač vratili još jednom. Utešno je jedino što su cena dinamičkih memorija stabilne već mesecima i što postoje dobri izgledi da se do kraja godine neće menjati.

Iako u ovim uslovima nije bilo nimalo jednostavno aktopili aranžman za organizovanu nabavku delova iz inostranstva — u trenutku kada svojim kupcima mara čipova ne gram strani trgovci nisu baš previše raspoloženi za naše devizne propise i štosove sa pekaricima od 1500 dinara — redakciji je ipak pošlo za rukom da postigne odgovarajući sporazum. Ovoga puta smo za opredelili za London iz dva razloga. Prvi je da se memorijski čipovi 4418 trenutno mogu naći jedino u Engleskoj (ne sumnjamo da će ih uskoro nelaviti i trgovci iz Austrije i Italije) a drugi je da je naš dopisnik Andelko Zgorelec obećao punu pomoć pa čak i kontrolu isporuke.

Iz organizacije akcije sa računarom galaksija mogli smo puno toga da naučimo ali ne žalost, malo šta možemo da promenimo. Redakcije naime nema nikakvog ulicaja na domaće privredne tokove a pogotovu ne na kretanje na svetskom tržištu čipova. Zato se naši uslovi verovatno neće dopasti konstruktorima koji gore od nestrpljenja da povećaju znanju svoje galaksije. Mi svega puta ne možemo da garantujemo ne samo cenu nego ni — rok isporuke. Možemo jedino da obećamo komponenta — štampana kola i čipove — i to samo za one koji pošalju narudžbenice do 10. avgusta. Tada će spiskovi biti zaključeni i na osnovu njih organizovana prelzvodnja štampanih kola odnosno nabavka čipova. Te praktično znači da isporuke ne može početi pre kraja septembra ali i to da će biti okončane u relativno kratkom roku. Naravno, ako nas i ovoga puta ne zatrpa 4000 narudžbenica kao za računar galaksija. Prilikom isporuke će biti poštovan redosled prijavljivanja — ovoga puta nije bilo preliminarnih narudžbenica pa ne može doći ni do zabune. Ljubitelji računara imaju gorka iskustva iz januarske akcije da je jeden den razlika u slanju narudžbenice značio i do tri meseca čekanja više.

Čitaoci koji su dobro upoznali mehanizme akcije sa računarom galaksija — znamo da su oni mimo naše volje i uticaje mnogima dobro zamlsli živca — primetiće dve novine znatno nepovoljnije ali realnije rokove isporuke i nešto drugačiju organizaciju isporuke štampanih kola vršiće se najverovatnije iz redakcije Galaksije. Time ćemo postići mnogo bolji uvid u njenu dinamiku a čitaoci će lakše doći do potrebnih obaveštenja. Isporuka čipova biće kad i do sada vršena iz inostranstva.

Cene koje navodimo za štampana kola i komplete čipove nisu razume sa konačne. To međutim ne bi trebalo previše da brine ljubitelje računara. Tačne cene za štampane kole objavićemo pelnaestak dana pre početka isporuke. Oni kojima to bude previsa skupo moći će da se predomisle i otkažu narudžbinu. Tačna cena kompleta čipova biće navedena u računu koji će svaki interesent dobiti iz inostranstva. Isporuka čipova biće vršena u skladu sa našim carinskim propisima.

Nijedan časopis na može da prauzme na sebe sve konstruktorske probleme svojih čitalaca a većine nejčešće nije raspoložene da preuzme nijedan. Proveren projekat, zaista predstavlja najveću vrednost koja za uopšte može ponuditi čitaocima. U našim uslovima, međutim važe nešto drugačiji zakoni — proverent projekti ne znače ništa ako se ne obezbede uslovi za njihovu realizaciju. Računari galaksija bi bez obzira na sve svoje kvalitete najverovatnije ostao samo mrtvo slovo na papiru da nije organizovan servis za nabavku delova. U njemu nije uvek svi funkcionisalo kako treba — rokovi su se razvlačili kao gladne godine a cene skakale kada smo se tome najmanje nadali — ali sada ipak ima u poganu najmanje 1500 galaksija i svakoga dana ih je sve više. Redakcija se još uvek nije umorila balansirajući između vetra svojih čitalaca i vetra svojih seradnika u ovoj akciji. Nadamo se da niste ni vi. Strpljenje je uostalom majka mudrosti! Ako nam ono ne pomogne, zaista ne znamo šta će.

Koliko šta košta

1 Štampano kolo za memorijsko proširenje (dvostruka štampa na vitroplastuj za konektorskom pločicom oko 1300 dinara

2 Štampano kolo za programator EPROM-a (jednostruka štampa na vitroplastuj oko 1000 dinara

3. Komplet čipova za memorijsko proširenje 48 K (čipovi podnožje konektor poštarine)

48 K	oko 40 funti
32 K	oko 30 funti
16 K	oko 20 funti

4. Komplet delova za programator EPROM-a (integrisana kola podnožje od 40 nožica i ZIF podnožje od 28 nožica, konektor)

oko 35 funti

Vazi do 10. avgusta

Narudžbenica
(štampana kola)

Molim vas da mi pošaljete pouzećem
1) štampane kolo za memorijsko proširenje (vitroplasti dvostruka štampaj za konektorskom pločicom **po orijentacionej ceni od 1300 dinara**
2) štampano kolo za programator EPROM-a (vitroplasti jednostruka štampaj po orijentacionej ceni od 1000 dinara

Mesto

Ulica i

broj

Ako mi konačna cene ne bude odgovarale zadržavam pravo da otkažem svoju narudžbinu deset dana pre početka isporuke. Štampana kola ću platiti poštaru prilikom preuzimanja vaše pošiljke. Narudžbenicu (original ili fotokopiju) dostaviti na adresu: Redakcije Galaksije 11000 Beograd Guševa vojvode Mišića 17 ZA STAMPANA KOLA

Narudžbenica
(komplet integrisanih kola)

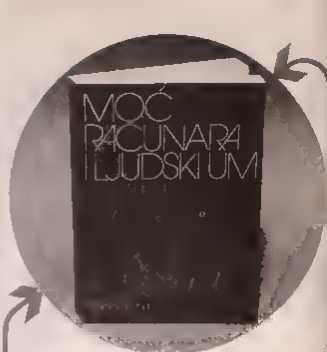
Molim vas da mi pošaljete predračun sa uslovima isporuke za sledeći komplet delova
- memorija za galaksiju 16 K
- memorija za galaksiju 32 K
- memorija za galaksiju 48 K
- programator EPROM-a za galaksiju
- programator EPROM a za ZX Spectrum

Mesto

Ulica i

broj

Narudžbenicu (original ili fotokopiju) dostaviti na adresu INTRAPRESS 19 Garway Road LONDON W2 4NH England



Džozef Vajzenbaum:

MOĆ RAČUNARA I LJUDSKI UM

Knjiga koja na izuzetan način osvetljava neke aspekte jednog od najsloženijih fenomena savremena tehnologije — **kompjuterizacija**. Pisana popularno i duhovito, ali bez naružavanja tehničke i naučne discipline poseduje i tu nesvakidašnju vrednost što pruža odgovore na neka često napoznatije problame a istraživačima postavlja nezaobilazna pitanja.

Formal 14×20 Str. 307 Broširano Lalinica
Cena 350-

RAD — IZDAVAČKA RADNA ORGANIZACIJA
11000 Beograd Moše Pijade 12 Tel 404 765 i 422 517

NARUDŽBENICA — RAČ/105

Ovim neopozivo naručujem knjigu
MOĆ RAČUNARA I LJUDSKI UM,
po ceni od 350.
Knjigu ću uplatiti **pouzećem** prilikom preuzimanje od pošta

(Poručilac)

(Broj pošte mesto

i adresa stana)

(Datum) (Potpis poručioca)

RAD

poziv na pretplatu

hemijsko tehnološki priručnik

(u 6 knjiga)



HEMIJSKO-TEHNOLOŠKI PRIRUČNIK pokriva specifične područja hemije i hemijske tehnologije. Zahvaljujući koncepciji svog višetomnog izdanja sve pojedinačne knjige čine celinu čime poprimaju i ranu vrednost i postaju stalni saradnik hemičara u njihovom radu

Prva knjiga

HEMIJSKI I FIZICKI PODACI I VELIČINE

Autori M Kramarević M Obradović B Dardavić G Dimić P Tadić D Šapa P Šaljević K Bjelajić N Radusavić J Ćirić B Miodanović V Prašić O Tatić-Janjić S Đorđević V Veljanović B Šerbanović Z Dizdar **Redaktor** Z Dizdar

Knjiga sadrži ujl izbor podataka z više osnovnih oblasti koja se najčešće koriste u eksperimentalnoj istraživačkoj i industrijskoj organizaciji. Data su numerička vrednost veličina na koja se odnose na fizičko hemijske osobine elementa jedinjenja i tehničkih materijala.

Iz sadržaja

- Matematike
- Definicije pojmova i jedinačini fizičke hemije
- Medunarodni sistem memih jedinica i preporuke
- Fizičke osobine elemenata i jedinjenja
- Fizičkohemijske tablice
- Osobine tehničkih materijala
- Opšti trivijalni i komercijalni oazivi materijala i hemikalija
- Podaci za proračun doza i zaštite od zračenja radioaktivnih izvora

Druga knjiga

ANALITIKA

Autori J Ćirić D Stojanović V Vajgand D Pešić J Cvetković V Antonijević B Kolšić M Krstanov D Hajduković Ž Knežević A Gegaik **Redaktor** D Ćirić

Ova knjiga je obrađena i sistemi zovana tako da daje na samo teorijske osnove na kojima počiva analitika ach analitička hemija nego i praktične upute i preporuke. Obrada je na osnovna klasična i savremena metoda organska i neorganska analiza. Materijal je sistematizovan veliki deo analitičkih podataka koji su neophodni za svakodnevna izvođenje pojedinih analiza.

Iz sadržaja

- Ravnoteže u rastvorima elektrolita
- Kvalitativna neorganska analiza
- Gravimetrija i volumetrija
- Elektroanalitičke metode
- Spektrohemijska analiza
- Radio-hemijske analiza
- Organske analiza
- Spektrografska analiza organskih jedinjenja

- Knjige će izaći iz štampe sukcesivno do kraja 1984
- Mogu se otpisati u najviše 10 mesečnih rata. Po izlasku iz štampe cena će biti znatno veća
- Pretplatnicima koji odjednom uplate ceo iznos pretplate odobravamo POPUST 20%

Treće knjiga

FIZIČKOHEMIJSKE METODE

Autori B Đorđević I Daraišlovačk D Pešić A Gegaik M Marinković M Pavićević A Mak B Ridnikar A Antić Jovanović M Jaćimović N Jakovljević Hajj M Rajčić I Draganić Z Dragenić K Zmudžev G Hajdukov N Cvetković D Cvetković **Redaktor** Đorđević I D Pešić

Knjiga tretira fizičkohemijske metode merenja koja se najčešće susreću u praksi

Iz sadržaja

- Optičke fizičkohemijske metode
- Elektrohemijske metode
- Optičke metode
- Razne metode

Četvrta knjiga

ŠTETNE I OPASNE MATERIJE

Autori O Stojanović N Stajanović D Kosanović **Redaktor** O Stojanović

Knjiga sadrži tabelarni pregled oko 3 000 raznovrsnih hemikalija sa zapaljivim, eksplozivnim i toksičnim karakteristikama kao i praćenim opšte opasnosti i rukovanja sa svima. Pored ovih karakteristika navedena su i određena najpovoljnija fizička sredstva za iste jedinjenja kao što su temperatura topljenja i ključanja gustina indeks refrakcije napon pare i gustina para

Iz sadržaja

- Direktno dejstvo štetnih materija
- Indirektno dejstvo štetnih materija
- Transport opasnih materija
- Skladištenje i pakovanje opasnih materija
- Rad sa opasnim materijama i štetnim
- Mere zaštite na radu
- Opasne otpadne materija
- Uputstvo za prvu pomoć
- Tabelarni pregled karakteristika opasnih materija

RAD — IZDAVAČKA RADNA ORGANIZACIJA
11000 Beograd Moše Pijade 12
tel 404 785 i 422-517

NARUDŽBENICA — RAČUNARI/103

Ovim se naručuje/u primjerak/a pretplatnom na
HEMIJSKO-TEHNOLOŠKI PRIRUČNIK (u 6 knjiga)
po pretpl cenl od 12.600.-
Uplatu ću izvršiti
GOTOVINOM – Po prijemu Radova uplatnica se **POPUSTOM** od 20% (10 080 din./
U ..... RATA – Uplatu ću (najviše deset) po prijemu uplatnica od RADA
(Nepotrebno precrtati)

U slučaju spora sadležan je Peti opštinski sud u Beogradu

|Potpis|

|Poštanska|

|Broj pošte mesto|

|i adresa stanal|

M P

(Datum) |Potpis naručioca|

Peta knjiga

HEMIJSKO INŽENJERSTVO

Autori G Popović B Đorđević V Valant B Šerbanović F Zdanski G Vunjak-Novaković S Končar-Đorđević Ž Grbavčić A Tesić A Mrajčev S Jakšimović Tjapkin S Pajanović A Tunović D Simonović D2 Hadžimanjlović A Tešić A Grozo R Radosavljević M Milinović B Pribićević S Dujković P Putanov M Ganeva G Branković A Dedijer Lj Milosavljević M Jovičić Z Popović **Redaktori:** S Končar-Đorđević D Simonović J Micić

U ovoj knjizi dati su teorijski osnovi kao i podaci koji se koriste prilikom rada sa tehnološkim i drugim projektantskim gasovima. Dati su i analitički kriterijumi i calkulacioni dati su primeri koji bolja ilustruju izneti materijal

Iz sadržaja

- Primenjena matematika u hemijskom inženjerstvu
- Termodinamika
- Principi i projektovanje tehnoloških operacija

A Operacije prenosa količine kretanja
Fluidi u mirovanju Fluidi u kretanju Strujanja u cevima. Mešanje Disperzivni sistemi Disperzije tela

B Operacije mehaničkog tretiranja čvrstih materija
Transport čvrstog materijala Sitnjenje materijala. Presejavanje

C Operacije prenosa toplote
Konvekcija toplote Zračenje toplote Toplotni bilans/ Ceni razmenjivači toplote Ukuvavanje Kondenzacija i kondenzatori Rashladni prenosi Gorivo i sagorevanje

D Operacije prenosa mase
Centrifugiranje prenosa mase Kontaktni uređaji za prenos mase sistema gas tečnost Apsorpcija Destilacija Ekstrakcija tečno tečno Luženje Pakovanje i kondicioniranje Sušenje i sušnice. Apsorpcija i jonska razmena Kristalizacija Dalalo difuzno operacija

- Osnovi teorije i projektovanja hemijskih reaktora
- Katalize i katalizatori
- Dinamika sistema, instrumentacija i automatska regulacija u hemijskom inženjerstvu
- Osnove projektovanja proizvodnih sistema hemijske (procesne) industrije
- Hemijsko-inženjerska ekonomika

Šesta knjiga

KOROZIJA I ZAŠTITA MATERIJALA

Autori B Mladenović M Petrović G Rikovski **Redaktor** O Tatić-Janjić

Knjiga obrađuje teorijske osnove korozije i tehničko-tehnološke principa zaštite materijala.

Iz sadržaja

- Elektrohemijska i hemijska korozija metala i legura
- Korozija materijala neorganskog i organskog porekla
- Korozione agresivnost raznih sredine
- Zaštita materijala od korozije

Knjige su štampane tajlinom formata 15 5x22 5 cm povezane u celo platno sa zlatotiskom na oko 5 000 strane

Cena u pretpletl 12.600.-

rad
1949
1984

Majstorije na računaru

softverska veza

Struje i kanali — najmoćnije spectrumovo tajno oružje

Dobro je o periferijskom uređaju razmišljati tako što će se on podeliti u dva dela: jedan odgovara softveru a drugi hardveru koji prima ili generiše podatke. Kod ZX bejzika softverska komponenta naziva se „struja", a hardverska komponenta „kanal". Ključna razlika je u tome što je struje bezličan protok podataka u program ili iz njega, dok kanal odgovara određenom uređaju, kao što je, na primer, ZX štampač. Struju treba shvatiti kao zbir podataka na njihovom putu ka nekom komadu hardvera ili od njega.



Struje . . .

Struje je određena nekim brojem od 0 do 15, njena osnovna operacija su čitanja i pisanja podataka. Naredba

INPUT#s ; ulazna lista

čitaće podatke iz struje „s" i prebacivati ih u promenljivu koju smo nazvali „ulazna lista". Na primer

INPUT#0; A, S; A$

čitaće podatke iz struje 0 i skladištiti ih u promenljive A, S i A$. Na isti način, naredba

PRINT #s; „print lista"

slaće podatke strujom „s" iz promenljivih u „print listi". Na primer

PRINT#0; TOTAL; A$

Obratite pažnju na to da se i INPUT i PRINT mogu koristiti na potpuno isti način kao i obična naredbe INPUT i PRINT. Svaki parametar koji se može koristiti kao i deo PRINT i INPUT može se uključiti i kao deo struje. Na primer, u važnosti su obe donje linije

PRINT#2, „RAČUNAR"; TAB (10); „GALAKSIJA"

INPUT#0, „Keko se zoveš?"; N$

Naredba PRINT šalje strujom 2 slovni niz „RAČUNAR", zatim nakoliko praznina i, najzad, slovni niz „GALAKSIJA". Obratite pažnju na to da podatak koji je poslat struji tačno odgovara podatku koji bi bio poslat ekranu. Naredbe INPUT je nešto složenije utoliko što ne sahteva samo podatak iz struje 0, već i šalje podatak u obliku slovnog niza „Kako se zoveš?". Svaki broj struje je, u stvari, vezan se dve struje podataka: jednu ulaznu i jednu izlaznu struju. Podatak upisan u struji bilo naredbom PRINT ili naredbom INPUT, šalje se izlaznom delu struje, a svaki podatak koji se čita iz struje dobija se iz njenog ulaznog dela.

Sa „narazvijenim" (tj. bez mikrodrajva) Spectrumom, mogu se koristiti još jedino naredbe LIST i iNKEY$. Potpuni oblik naredbe LIST je

LIST #s; n

gde je „s" broj struje kojom se program može izlistati, a „n" je broj linije od koje počinje listanje. Na primer

LIST 1

izlistaće ceo program pomoću struje 1.

Druga naredbe orijentisana ka struji, iNKEY$, može se koristiti za prihvatanje pojedinačnog bajta iz bilo koje struje koje je vezana sa uređajem koji podržava ulaz. Naredbe

INKEY$ s

vratiće kod jednog karaktera iz struje „s".

. . . i kanali

Lako je shvatiti ideju o struji podataka, ali se postavlja pitanje kako se određena struja povezuje sa određenim periferijskim uređajem. Odgovor je: da bi se neki podatak poslao ili primio određenom strujom, ona more biti najpre otvorena. Otvaranje struje ima dva cilja: ono povezuje potrebnu struju sa određenim periferijskim uređajem i, u isto vreme, signalizira koji će uređaj biti upotrebljen. Često, uz signaliziranje da će jedan uređaj neposredno biti korišćen, otvaranje struje obuhvata i uputstvo instalisanom uređaju kako da dođe u stanje u koma će se koristiti; ovo upućivanje, međutim, vrlo mnogo zavisi od samog uređaja.

Otvaranje struje ostvaruje se naredbom

OPEN#s c,

gde je „s" broj struje koja se otvara, a „c" slovni niz koji određuje kanal sa kojim je ona povezana. Prema toj naredbi, odredište svakog podatka poslatog strujom „s", biće kanal „c", koji će, isto tako, biti i izvor svakog podatka pročitanog iz te struje. Pre nego što se može dati preklican primer korišćenje naredbe OPEN, treba znati kojim kanalima SPECTRUM raspolaže.

Spectrum bez mikrobrojeva raspoznaje samo tri raličita kanala

K — kanal tastature

S — kanal ekrana

P — kanal štampača

Tako

OPEN #5, „K"

otvara struju 5 i vezuje je sa tastaturom. Na osnovu naredbe

INPUT#5 A, S

dobiće se podaci sa tastature na isti način kao i pri normalnoj naredbi INPUT. Međutim naredba

PRINT#5; „GALAKSIJA"

šalje sada podatke izlaznoj strani struje 5, koja je povezana sa donjim ekrana. Na taj način, slovni niz „GALAKSIJA" upisuje se u donji deo ekrana koji je normalno rezervisan sa INPUT poruke. Ako to pokušate, nije verovatno da ćete moći videti niz, jer se donji deo ekrana briše čim se program zaustavi ili čim računar naide na naredbu INPUT. Ako hoćete da vidite efekat slanja podataka u „ulaznu oblast" ekrana, pokušajte sa

10 OPEN#5; „K"
20 PRINT#5; RND
30 GOTO 20

Videćete na ekranu nasumična brojeva, koji beže odozdo naviše. Program će se završiti sa porukom o greški OUT OF SCREEN.

Prvi put kada požalite da na svoj Spectrum vežete neki periferijski uređaj, sudarićete se sa problemom kako mu slati ili kako od njega primati podatke. Obično se misli da je lakše konstruisati hardversku vezu sa Spectrumom nego „softversku vezu" sa ZX bejzikom. Međutim, Spectrum ima veoma elastične i sofisticaran metod rukovanja periferijskim uređajima, zasnovan na korišćenju „struja" i „kanala", pa je razmerno lako povezati periferijske uređaje sa ZX bejzikom tako da ih računar tretira kao svoj sopstveni hardver. Koristeći veoma jednostavan solver u mašinskom jeziku moguće je slati podatke svakom uređaju naredbama PRINT i LIST i primati podatke od svakog uređaja naredbama INPUT i INKEY. To, praktično, znači da se pirmanjeni programi mogu pisati u standardnom bejziku bez prečestog pozivanja mašinskog koda. Čudno je da se, uprkos tome što standardni Spectrum koristi tako elastičnu tahniku, ona jedva i pominje u priručniku!

Mapa kanalskih zapisa

kanal za tastaturu
CHANS— *adresa prnnt rutine za donji deo ekrana*
+2 — *adresa input rutine za tastaturu*
+4 — *K kod kanala K*
kanal za ekran
+5 *adresa print rutine za ekran*
+7 — *adresa rutine za tretiranje grešaka*
+9 — *S kod kanala S*
kanal sa redaktoraki bafer
+10 — *adresa bafera input rutine*
+12 — *adresa rutina za tretiranje grešaka*
+14 — *R kod kanala R*
kanal za ZX printer
+15 — *adresa rutina za ZX printer*
+17 — *adresa rutina za tretiranje grešaka*
+19 — *P kod kanala P*

Mada u principu svaka struje ima ulaznu izlaznu stranu, u praksi jedini kanal koji može da primi r uiaz I Izlaz jesta kanal tastatura Druga dva kanale ekrena i štampača samo su izlazni kanaii i svaki pokušaj da se sa njih očitavaju podaci završiće prijavijivenjem greške J. Treba primetiti da je ovo ograničenje iskijočivo karakteristika hardvera za koji je vezana struje

Za jedan kanal je moguće vezati više od jedne struje, eli eko se želi promeniti kanal sa kojim je struja vezana veza se najpre mora ukloniti sa CLOSE. Dakle:

CLOSE # s

ukloniće svaku postojeću vezu izmedu etruje „s" i kanala U ovom smielu. CLOSE je za kanal suprotno od OPEN. Zatvaranje etruje može se koristiti i zato da se komponente hardvera jednog kanala obavesti da strujl više nije potrabna

Formati kanala

Četiri struje, 0 do 3, automatskl se otvaraju prilikom inicijalizacije računara

struje	kanal
0	K
1	K
2	S
3	P

Tako će, čak I bez OPEN, naredba

PRINT # 2, „GALAKSIJA "

episati tekst na ekranu Spectrum koristi ove etruje za upućivanje programskih podataka u odgovarajući uređaj Na primar, „PRINT šalje podatka ka strujl 3. Ovl odnosi struja r kanala mogu sa pormeniti korišćenjem naredbe OPEN, aii sama etruje se na mogu zatvoriti Pokušaj da sa zatvori jedna od ovih etruje završave njenim ponovnim otvaranjem ka evom prvobitnom kanalu.

Ako struje I kanali treba da se koriste za povezivanje dodatnih uređaje, tada ja očigiedno važno znati unutrašnje formate koji su korišćeni za skladištenje informacije o kanaiima, kao i koji je kanal vezan sa kojom strujom

Informacija koje definiše svaki kanal uskladištena je u zoni kanalskih informacije položeći od CHANS i završavajući sa PROG-2 (gde su i CHANS i PROG sistemeke promenijive). Svaki kanal ima poseban „kanalni zapis" sledećeg formata

adresa veličina
n 2 bajta adresa izlaznog programa
n+2 2 bajta adresa ulaznog programa
n+ 1 bajt kod kanala

gde su ulazni I izlazni programi sabrutina mašinskog koda izlazni program mora da prihvati Spectrumove kodove karaktera koji mu se „doturaju" preko A registra. Ulazni program mora da vrati podatke u obliku Spectrumovih kodova karektera i da signalizira de su podaci dostupni setovanjem tzv flega za pranos (carry tiog) Ako nikakvi podaci nisu dostupni, to se signalizira resetovanjem Z I C flega Ako kanal na može da prihvati podatak sa ujaza iii izlaza, o tome traba da bude obavešten program koji trstira greške Standardni način tretiranje grešaka u ZX bejziku jeste putem jednog restart poziva na adresi 8. Na mašinskom jaziku to izgieda ovako

ERROR RST 0008
DEFB errocode

gde je „errocode" numerički kod poruke o grešcr koje se daje korisniku.

Kanalni zapisi za tri standardna Spectrumova kanala i jedan dodatni kanal koji do sada nije opisan prikazani eu u Tebeli 1.

Obrailte pažnju na to da je svaki kanalni zapis standardnog formata I da jedino K kanal može da prihvati I ulaz i izlaz Novi, R kanal Spectrum koristi interno za slanje podataka redaktorakom baferu Naredba OPEN neće dozvoliti korisniku da poveže struju sa R kanaiom, pa je tako njegovo korišćenje ograničeno

Važna primedba: format jednog kanalnog zapise drukčiji je kada se koriste mikrodrajv i interfejsi

Tablica struja

Veza etruje sa kanalima uskladistena je u tablici struje, u oblasti sistemskih promennjivih, na prostoru od 38 bajta položeći od STRMS (edresa 23568) Tablica struja I svaki par bajtova u toj teblici nose jedan broj „x" koji predstavija edrsu početka jednog kanalnog zapise Namesto prostog skladištenja adresa kanalnog zapisa, „x" je „rastojenje" na kome sa kanalni zapis naiezi od početka zone kanalskih informacije

početna edresa = edresa kanala + x − 1
zona kanaiskih zapisa

Pošfo postoji maksimaino 16 struja, moglo bi se pomisliti da su dovoljna 32 bajta (tj jedna kanaina adresa po etruji) za skladištenje svih asocijelecije kanala i etruja. U stvari, dodatnih 6 bajta koriste se za skladištenje kanainih informacija za tri unutrašnje struje koje odgovaraju brojevima struje 255 254 I 253 Te tri unutrašnje etruja au automatski povezana se kanaiima R, S, odnosno K i nisu dostupne iz bejzika Međutim, prisustvo te tri unutrašnja etruje mora se uzsti u obzir kad se pokušava naći adresa kanale kojl odgovara na kojoj od atruje 0 do 15 Prva tri uiaza u tablicu struje odnose se na unutrašnje etruje 253 do 255 a četvrti ulaz daje edrsau kanala koji će se koristiti se strujom 0 idt To znači da je početak tablice struja, što sa spoijnih etruje tiče na adresi

STRMS+6 iii 23574

Adresa kanala povezanog sa strujom e (a je u okviru 0 do 15) skladišti se u dva memorijska mesta sa početkom na

23574+S*2

Kada se jedna etruja otvori za jedan određeni kanei, ispituja se tablice struje da bi se naša adresa kanale Kad je etruja zatvorena, njena edresa za uiazak u tablicu etruje podešena je na nulu Tako se nuiti ulazak u tablici etruje koristi za otkrivanje pokušaje de se koristi atruje koja još nije bila otvorena.

Ovaj eistem kanainih zapisa i tablice struja proširen je i na korišćenje mikrodrajva, aii njegove bitna karekterlstika ostaju iste Svaki kanai je opisan kanainim zapisom, a etruje su povezana sa kanainim zapisima pomoću tablice struje.

Pre nego što predemo na razmatranje kako praktično iskoristiti poznavanje struje i kanala, vredi pomenuti jednu sistemsku promenljivu CURCHL. U njoj se čuva adrese tekućeg kanala. Na taj način, nakon naredbe kao što je PRINT # S CURCHL sedrži adresu kanala povezanog sa strujom „s "

Kanali po meri

Bejzik naredbe iN i OUT predstavijeju najjednostavnlji način za slanje, odnosno primanje podataka direktno sa odgovarajućeg ulazno-izlaznog porta Na primar ako ja generatoru zvuka dodeljen port 31 se kontrolu njegove grekvencije, tada će

OUT 31 t

slati podatka (t= 0 do 255) koji će direktno uticati na visinu tona. Za jednostavne uređaje i uređaje koji se kontrolišu pojedinačnim bitovima, iN i OUT au vrlo pogodna naredbe Međutim, ako je uređaj „znakovno orijentisan", tj eko prima i odašilje podatke u obliku znakova, tada eu iN I OUT nepodesni Na primer, paraielni štampač i modem su znakovno orijentisani uređaji, I najpode-

```
PROGRAM 1


 10 DATA 01,00,254 237,121,201
 20 FOR a=23297 TO 23301
 30 READ d
 40 POKE a,d
 50 NEXT a
100 GO SUB 1000
110 FOR i=0 TO 7
120 LPRINT i
130 NEXT i
140 GO TO 110
1000 LET c=PEEK 23631+256*PEEK 2
3632
1010 LET c=c+15
1020 POKE c,23296-INT (23296/256
)
1030 POKE c+1,INT (23296/256)
1040 RETURN
```

snijі način da se se njima radi jeste putem običnih PRINT i INPUT naredbi Jasno je da najbоlji put vodi preko struja i kanala.

Uključivanje novog periferijskog uređaja u sistem struja i kanala može es ostvariti na jedan od dva načina: ili promenom adresa uskladištenih u nekom postojećem kanalnom zapisu ih stvaranjem jednog potpuno novog kanalnog zapisa

Prvi metod se sastoji u ubacivanju (POKE) novih adresa u postojeći kanalni zapis Kanal će, tako, biti usmeran na vaše sopstvene programe na mašinskom jeziku, smeštene negde u memoriji Na primer, pretpostavimo da želite da povežete etandardni štampač umesto ZX štampača Umesto postojeće adrese na prva dva mesta kanalnog zapisa ZX štampače, (CHANS + 15), treba da postavite adresu sopstvenog programa za pogon štampaća. I naredbe LPRINT i LLIST, kao i evaka druga naredba koje es odnosi na struje otvorene ka kanalu P, slaće podatke novom štampaču

Pisanje novog programe za upravljenje štampaćem je u principu, tako sve što treba da es uradi jeste dovesti ASCII kodove u A registar i posleti ih u štampač Medutim, Spectrumov set znakova obuhvata i mnoge kodove koje standardni ASCII set karaktera ne poznaje. Pre nego što se pošalju preko štampača ovs kodove treba konvertovati Na primer evi kontrolni kodovi unuter PRINT naredbe biće posleti preme Dodatku A Spectrumovog priručnika. To znači da LPRINT TAB (10) šalja ASCII kodove 23, 10 i 0; 23 je Spectrumov kontrolni kod za TAB a sledeća dva kode su najmanji i najvažniji bajt parametra TAB funkcije Ovi kodovi, medutim, za štampač mogu da znače nešto sasvim drugo. Isto tako, koodovi za bajzik naredbe ne znače za štampač ame baš ništa Oni se, zato, moraju prevesti u niz ASCII znakova, tako de es postigna efekat kao kada es evaka naredba kuca slovo po slovo.

Primer ovog metoda povezivanja nakog novog periferijekog uredaje predstevija stedeći program koji umesto izlazna adrese uskladištena u P kanalu postavlja adresu mašinakog programa koji je uskladišten u baferu za štampač Ovaj sistem funkcioniše samo etoga što ja ZX štempač van upotrebe!). Nov izlazni program na radi ništa korisno es podacima, osim što ih šalje ka portu 254 koji kontroliše zvučnik i ivičnu

```
Tabela 2

23296  outdir   LD   BC, 254   — unesi u BC registar 254
23299           OUT  (C), A    — pošalji sadržaj registra
                               A na port 254
23301           RET            — vrati se na bajzik
```

Jednostavni program za opsluživanje periferijskih uredaja u mašinskom jeziku

boju, to obezbeduje da se njegovi efekti vide i čuju. Rutina kojom se postiže ovi afekti data je u tebeli 2. Ona je korišćena u bajzik listingu prikazanom u Programu 1.

Mašinaki kod izlaznog programa skladišti es u DATA naredbi u liniji 10 i unosi u memoriju pomoću linija 20 do 50 (23296 = početna adresa bafera za štampač). Subrutina 1000 menja adresu u kanalnom zapisu za kanal P. Linija 1000 dobija edresu početka kanalna oblasti u c (CHANS), a zatim linija 1010 nalazi početak kanalnog zapisa za kanal P(CHANS + 15). Linije 1020 i 1000 ubacuju (POKE) adresu novog izlaznog programe u prva dva mesta u kanalnom zapisu. Ako startujete ovaj program, videćete ivični biesak i veoma diviju promenu. Ako zaustavite program, dobićete dalji dokaz da novi izlazni program šalja podatke kontrolnom portu za border i izlistavanjem (LLIST) programa (Pažnje: pre primena ovog programe treba odvojiti ZX štampač)

Novi kanali

Menjanje adresa uskladištenih u postojećim kanalnim zapisime predstavlja lak metod za povezivanje novih uredaja at, na žalost, on onemogućuje jeden od Spectrumovih postojećih I/O uredaja U praksi je nemoguće moditikovati kanal K, jer se njegove adrese ponovno uspostavljaju kad god se izvrši naredba INPUT. Kanali S i P ostaju, tako, kao jedini kandidati za modifikaciju. Pošto je kanal S isuviše dragocen, jedini stvarni kandidat je P. To je lepo sve dok ne osetita potrabu da istovremeno koristite više od jednog periferijekog uredaja

Da bi se omogućilo istovremeno povezivanje nekoliko periferijskih uredaja, potrebno je uvesti nove kanale. Dodevanja jednog novog kanata izgleda vrlo teko, eli treba uzeti u obzir nekoliko sitnijih detalja. Pre svega, kanalni zapis je moguće stvoriti bilo gde u memoriji, na esmo u zoni kanalskih informacije, medutim, ako je kanalni zapis usktedišten iznad INPUT redne oblasti (počinje kod WORKSP), aistemska promenljiva CURCHL u kojoj es čuva adresa tekućeg kanala biće izmenjena nekon evake naredbe INPUT. To naravno znači da će edresa kanala biti izgubljena i Spectrum će otkazati. Medutim, ako je kanalni zapis uskladišten ispod INPUT radna oblasti, tade sve funkcioniše kako treba. U našem primeru korišćen je bafer štampača za skladištenje novog kanainog zapisa i novih izlazno-ulaznih rutine. Druga teškoće je u tome što naredbe OPEN i CLOSE funkcionišu samo sa standardnim kanatnim zapisime za K, S i P, to znači da je potrebno obezbediti i potprogram za povezivanje kanale es nekom strujom i, ako je potrebno potprogram za njegovo zatvaranje. Sve to zajedno obezbeduje mašinska rutine prikazana u Tebeli 3

Prvih pet bajtova čina nov kanalni zapis. Program koji počinje od adrese 23301 je izlazni program i on jednostavno... šalje kod u registru A u izlazni port 254 — port zvučnika i ivična boje. Program koji počinje od adrese 23307 je ulazni program — on obaveštava o greški da bi ukazno da se ovim kanalom ulaz nije dozvoljen. Naravno u stvarnoj primeni svaki od ovih programa može da bude mnogo složeniji. Program 2 predstavtja bajzik program za unošenje i ektivirenje mašinske rutine

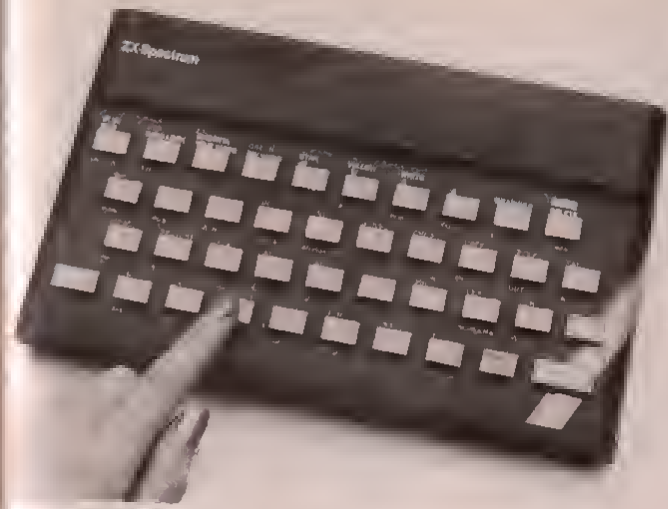
Linije 10 i 50 unoes nov kanalni zapis i ulazno-izlazne rutina u bafer štampača. Potprogrem 1000 otvora struju s za nov kanal. Drugim rečima, ona je ekvivalentna naredbi OPEN #, E". Linija 1000 izračunava tačnu adresu za struju s u teblici struje. Linije 1010 i 1020 izračunavaju tzv. ofset preskok za novi kanatni zapis a linija 1000 i 1040 skladište njegovu adresu u tebilci struja. Linije 100 koristi podprogram 1000 za otvarenje struje 5 ka uredaju E a

```
PROGRAM 2


  10 DATA 0,91 11,91,69,1 0,254,
237,121,201,207,18
  20 FOR a=23296 TO 23308
  30 READ d
  40 POKE a,d
  50 NEXT a
 100 LET s=5 GO SUB 1000
 110 PRINT # RND,
 120 GO TO 110
1000 LET a=23574+2*s
1010 LET c=PEEK 23631+256*PEEK 2
3632
1020 LET r=23296-c+1
1030 POKE c,r-INT (r/256)*256
1040 POKE a+1,INT (r/256)
1050 RETURN
```

```
23296  chanrac  DEFB  5          — bajt manje težine u izlaznoj adresi
23297           DEFB  91         — bajt veće težine u izlaznoj adresi
                                 (91×256+5=23301)
23298           DEFB  11         — bajt manje težine u ulaznoj adresi
23299           DEFB  91         — bajt veće težine u ulaznoj adresi
                                 (91×256+11=23307)
23300           DEFB  "E"        — kod novog kanala ("E"=69)
23301  outdriv  LD    BC, 254    — unesi u registar BC 254
23304           OUT   (C), A     — pošalji sadržaj akumulatora
                                 na port 254
23306           RET              — vrati se u bajzik
23307  indriv   RST   8          — pozovi rutinu za greške
                 DEFB  18         — kod za poruku „invalid device"
```

Rutine za uvođenje novog kanala i opsluživanje periferijskih uredaja preko njega



Pod velom tajni iako se smatra da se o njemu već odavno sve zna Spectrum ne prestaje da iznenađuje programere

...nije 110 do 120 šalju kodova koji odgovaraju nasumičnim brojevnama zvučnom r ...vičnom kontrolnom portu Zaustavljanjem programa i unošenjem naredbe LIST# 5 može se proizvesti i biesak boje i zvuka, što znači da je program iziisten ka portu 254! Ako promenite fmlju 110

110 INPUT # 5 I

dobićete poruku o grešci, što znači da se ...ej kanal na može koristiti za ulaz Naravno, umesto 254 može se postaviti adresa nekog štampača, programatore EPROM-a ili ganetore zvuka

Ovim što se moraju pisati obuhvatniji i specijelizovaniji programi nema nikakvih teškoća u dodevanju novih kanalnih zapisa. Jedino treba voditi računa da gornji program neće funkcionisati eko su spojeni mikrodrsjvovr

Ulazni kanal

O problemima pisanja izlaznog drajvere već je ranije bilo reči Na kraju, traba pomenuti dodatne zahteve za ulazni drajvar.. Ako jedan ulazni kanal treba, poput osmobitnog A-D kovartora, da prihvata pojedinačni znakovni kod tade je najboija bajzik naredba INKEY$ # koje će vratiti pojedinačni znak Međutim eko planirate da koristite INPUT da biste učitali čitav znakovni niz tade morate biti svesni dve stvari Prvo, naredba INPUT ima i izlaznu struju, i nije dovoljno zameniti izlazni program programom za tretiranje greške; mora se obraditi svakr podatak koji šalje naredba INPUT, čak i u slučaju da se ona prosto zanemaruje! Drugo, naredba INPUT prihvata podatke kao da su otkucani na tastaturi. To, prektično znači eko koristite INPUT # s B de biste učitali neki broj u promenljivu B, program mora da šalje niz ASCII kodova koji odgovaraju brojevima i de se završi se jednim ENTEH kodom, kao da broj ulazi sa tastature Na kraju, treba obratiti pažnju i na ovo čak kada se podaci očitavaju i sa specijelnog komada hardvara, naredba INPUT će tačno interpretirati sve kodove — ona uvek funkcioniše kao da prime struju znakovnrh kodova kojr odgovaraju dirkama koje su pritisnute na tasteturi

Spectrumov sistem struje i kanale predstavlja neočekivanu premiju za programera na ZX bajziku Korišćen unutar programe, on obezbeduje nezavisnost uređaj i uopšte povećanje elesiičnosti bez ikakvih negativnih strana Progremeru na mašinskom jeziku struje i kanali pružaju idealnu sistku za softverska interfejse se novim uredejima Primari dati u ćienku predstavljaju polaznu tačku za sva resina softverska interfajse koje biste možda želeli da napišeta

(Electronics end Computing Monthly)

Mali oglasi

• ***Spectrumavci*** *— veliki izbor najprograma po najnižoj ceni uz novi popusi hobi panaitator horace pssst i drugi Cena svih programa je 200 ND Tel 345-490*

• *Disk-drive za Appla (II II+ IIe) aatpuno nov prodajem Tel 011/4443-351*

softwara club
Garantujemo najniže cene u zamiji Bilo koji program po vašem izboru za 100 din — deset šal programa za samo 600 dinara
— Manic Minar
— Paneiralor
— Hobri
i mnogi drugi programi koji stižu iz Engleske Svi programi su snimljani i venikovani direkino iz Speciruma Pišita di se javila telalonom Andrić Dejan Danieova 64 11060 Beograd tel 011/764-041

• ***GALAKSIJA*** *— komplet čipova samo 7500 din (2K verzija) Svakih dodatnih 2K samo 1750 din Isporuka odmah SHARP 1500 i printer 80 000 din Sazović Jakov Pive Karamatijevića 31 Beograd Tel 011/496-685*

ZX SPECTRUM 16/48K
i dalja najatfiniji software u gradu SAMO 50 DINARA PO PROGRAMU Puno novih programa za 16 i 48K
Skripte za mašinac
Super brza usluga
Specijalni programi po porudžbini
TRAŽITE NOVI BESPLATNI KATALOG
Kosić Vladimir Varvarinska 35 11000 Beograd tel 011/400-823

COMMODORE 64
Programski jezici
kompajlari
asembleri
word procesori
najnovija igre
kućni programi
poslovni programi
muzički sintesajzeri
sekvenceri kompzari
edukalvni programi
kurs BASIC-a 130 KB
govor-voice generalor
super gralika
spreadsheeli
razmena programa
Vaselin Milosavljević 11000 Beograd Vitanovačka 42 stan 45 tel 011/452-653 (po podna)

i dalja sa Vama COMPUTERLAND
Naša specijalnost je ZX SPECTRUM
Sistemskrt programi Namenski progrl'ami igra
Basicnalan bllc-kalalog (na, progr r cena) Kialog sa opisma svih progr r cenama 100- din
— COMPUTERLAND J Gagarina 120/62 11070 Novi Beograd TEL (011) 162-774

ZX-Spectrum programi na kasetama Za 60 din nudim pascal forth davpac arcadia pe nairator hma-gate jet-pac flight simulation loolbal mai nagar 3D-tunnel hobbri cyrus chess manic minar monopoly kong mr wimpy pool paintbox ant attack hignoon bridge battle 1917 i jos 250 d rugih Za kalalog pošaljite 20 dinara Razmanyujem VHS-video maienjal Marko Marković Dž Bijedića 27/XI 71000 Sarajevo

Za comodore-64 VIC-20 preko 200 fabričkih programa prodajem Besplaten katalog? Moguća razmena? Jeftino? Andrija Kolundžić, Vojvode Brane 31/IV 11000 Beograd tel 011/424-325

Comet
SOFTWARE
Sve za ZX SPECTRUM
— video igra
— ...glčke igra
— nau...nski programi
— knjige
SPECIJALNO!!!
Ako vam trebaju usluga na printeru obratite nam se VRŠIMO
— programski listing
— asemblarski listing
— kopiranja crteža u vis...oj razol uclj...
Iskoristite LETNJI POPUST!
MILOVANOVIĆ LJUBIŠA
Petra Lakovića 57
1100-0 BEOGRAD
tel 01 1/558 007

Skriota na ZX81 i spectrum na srpskohrvatskom
1 basic (600 N din)
2 mašinski jezik (800 N. din.)
3 najnoviji i najnoviji fiskinzi programa iz stranih časopisa (600 N. din.) 50 str.
4 šeme i uputstva za hardver dodatke i samogradnji (A/D kovertor 1/5 port, RS232 interfejs light pen i mnogi drugi) na 50 strana (600 N din.)
Vlasnici kompjutera Commodore 64 i commodore VIC 20
javite na radi razmena programa i literature Kolundžić Andrija Vojvode Brane 31/4, Beograd, tel 011/424-325

škola akcionih igara

Igre na računaru

Cilj škole je da vas, kroz dosta primera, uvede u neke programerske veštine koje će vam pomoći da kreirate sopstvene akcione igre. Neki od primera programa predstavljaju interesantne igre koje dodatno ne mogu da se mere sa komercijalnim programima, ali koje obzirom da ćete ih sami otkucati i shvatiti njihovu realizaciju mogu da vam pruže mnogo časova zabave naročito eko spadate u one vlesnika računere koji ne žele samo de se igrelu

Svi znamo da su dobre akcione igre moguće samo na mašinskom jeziku e ipak smo ovu školu propretili bejzik primerima Zašto? Čini nam se, pre svega da je nedovoljan broj čitelaca ovih redove dobro upoznat sa mešinskim jezikom pa bi mašinski programi odvlačili pažnju od onoga što program treba de ilustruje Kade savladete mašinac, pogotovu eko posedujete knjigu Disassembllreni Spectrumov ROM neće vam biti teško de primere bukvalno prevedete na mašinski jezik ili da u programima koje ćete sastavljati koristite tehnike neučene u ovoj školi

Škole je u osnovi namenjena visanicima Spectruma koji će imeti privilegiju da sva dete primere Isprobaju na svom računaru bez ikakvih preprevki I za vlesnike drugih računera, međutim, može da bude korisno de je pročiteju tehnike koje ćemo objasniti su univerzalne dok naredba jesno mogu da budu izmenjena Dovoljno je de držite uputstvo za upotrebu vašeg računere na dohvatu ruke dok čitate ova redova i sigurno ćete sa sneći bez većih problema

„Pritisni bilo koji taster . . .“

Nakada su računeri bili namenjeni uglevnom obradi brojeva koji su se unosili primenom naredbi koje su vrlo slična dobro nam poznatoj naredbi INPUT Jesno je de na ovaj nečin (korisniku je ostavljano skoro beskonačno vreme za razmišljenja i prijem podeteka kvari izgled ekrana) ne mogu de se primetu podeci potrebni za ekcione igre pa je u bejzik uvedene naredba INKEY$ koje deje kôd pritisnutog testere Ako se na primer, u programu nalazi naredba 5 LET X$ = INKEY$ promenljivoj X$ će biti dodeljeno slovo a eko je igreč pritisnuo e, A ako je pritisnuo SHIFT A i tome slično Kod Spectrume je važno primetiti de naredba INKEY$ na registruje trenutek kade je tester pritisnut kao kod mnogih drugih kompjutera Ukoliko na primer računar naprekidno izvršava naredbu 5, e korisnik stalno drži tastar A pritisnut X će neprekidno dobijati vrednost „A“

Naredba INKEY$ i pored ogromnih prednosti nad INPUT nije pogodna za akcione igre Zašto? U mnogim igrama je neophodno da se korisniku omogući istovremeno pritiskanje dva ili više tastera (kretanje i pucanje, na primer). Zato Spectrum poseduje moćniju i na žalost komplikovaniju naredbu IN One ne deja kao rezultat alfanumertk nego broj Ove naredba osim toga na tastira jeden nego čitavih pet tastere u isto vreme, tih pet tastera ćemo nazivati blok

Svaki blok tastera ima svoj broj koji nalazimo u tabatl 1 i stevljamo iza IN naredba (eko na primer želimo da testiramo tastere 1 2 3, 4 i 5 koristićemo naredbu LET A = IN 63486) Ukoliko ni jeden od tastera iz bloka nije pritisnut IN će vretiti broj 255 (uz jednu malu ogradu eli o njoj nešto docnije) Svaki pritisnuti tester smanjuje ovej broj za 2 dignuto na stepen n, gde se n tekode čita iz tebele Neka su na primer pritisnuti tasleri 1 2 i 4 PRINT IN 63486 će tada dati broj 255−2⁰−2¹+2³=244 Oni melo bolje upozneti sa binernom logikom svakeko će zapaziti da je svakome tasteru dodeljen po jedan bit koji je setoven eko je taster otpuštan i resetovan eko je on pritisnut Zato je sasvim jednostavno ispiteti de li su na primer pritisnuti istovremeno testeri 1 i 2, pri čemu se na obraće pažnje na etanje preostale tri I naredba kojom se to postiže je

IF (IN 63486 AND 3) = 0 THEN PRINT „Prilisnuti su 1 i 2“

Čini nem se de će vam bili od koristi da uložiti odredent napor u razumevanja ove naredbe jer ćete eličine imati pritike de korietita u mnogim drugim prilikama. IN 63486 deje broj izmedu 0 i 255 koji možemo da zamisilimo kao niz od 8 binernih jedinios i nule Desna dva bite odgovaraju tasterime 1 i 2 respektivno ukoliko je neki od njih 0, odgovarajući taster je pritisnut Preostali bitovi, međutim, imaju sadržej koji zavisi od loga de li su tasteri 3 4 i 5 prilisnuli, što nas kao što rekosmo ne interesuje Treba dakle da pronademo način da zanemarimo vodeći 6 bitova dobijanog broja odnosno de ih pretvorimo u nule Upravo je tome namenjena konstrukcija AND 3 Zapazimo najpre da se broj 3 prikazuje binarno kao 00000011 Logičke funkcije AND (i) nalaže računaru de proredi brojeve IN 63486 i 00000011 bit po bit Ako su oba bita koja se porede jedinios rezultat će takoda biti 1 u protivnom rezultat je nule Obzirom da broj 3 ime šest vodećih nule prvih šest bitova razultate će takode biti nule što smo i želeli Posledna dva bite neće promeniti vrednost jer ih logički množimo (AND se poneked naziva logičko množenje a OR logičko sabiranje) jedinicom Ukoliko su tasteri 1 i 2 pritisnuli te dva bite (i samim tim čitav rezultel) će biti jednaki nuli dok će u protivnom rezultat uvek biti različit od nule Ovakvo „maskiranje“ je vrlo često u upotrebi pogotovu kada pišemo mašinske programe jer predstavlja najcelishodniji način izdvajanja nekoliko bitova i jednog bajt...

Kao primer svega izloženog, dajemo jednostavni program za pisanje i crtanje po ekranu leko to nije akciona igre u svakoj akcionoj igri more de se crta po ekrenu zavisno od onoga što korisnik kuca, pe će tehnike naućena analizom ovoga progreme svakako biti od koristi Primetite da se u programu ne više meste koristi naredba INKEY$ — pokazuje se da je one jednostavnije od IN pa zelo ovu drugu traba koristiti samo u slučajevima kade postoji mogućnost da su pritisnuta dva tastere istovremeno — u našem primeru u naredbame 50 i 60

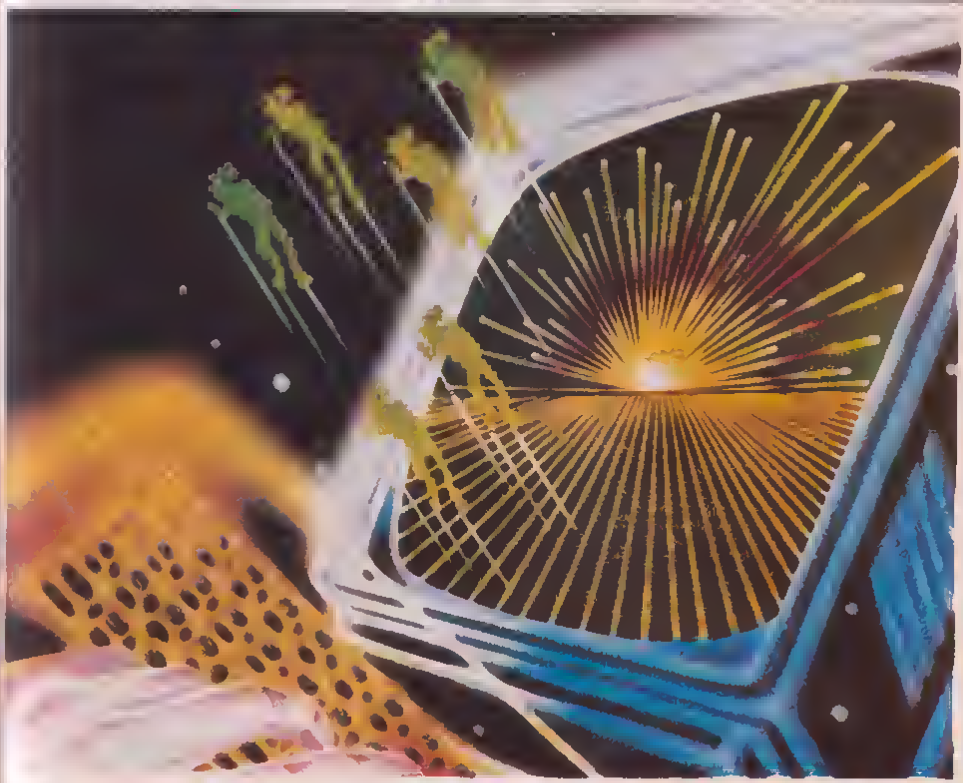
Obretimo najzad pažnju ne liniju 20 u kojoj se skriva naredba BEEP koja nerevno ne smete ali nije ni mnogo potrebna Došlo je dekle vreme da izlozimo ogradu koju smo pomenuli Spectrumov operativni sislam ne mora uvak brojeve pritisnutih tastera oduzimati od 255 — u upotrebi su i druge vrednosti U nekim slučajevima medutim, računar se sigurno vreća u normalizovano stenje i usvaje ovu konstentu Najjednostavniji recept za ovu normalizaciju je izvršiti BEEP pre ispitivanja testera pomoću IN Ovo objašnjenja morete, ne žalost de primite kao i mi — zdravo za gotovo Naredba IN jednoslevno ima mnoge druge namene osim ispitivanja tastalura pe se pojavljuju i neki artetakti

Marsovci dolaze

Sva ekciona igre mogu da se kreiraju korišćenjem naredbe PLOT tj crtenjem figure svaki put kad se ukaže potreba tečku po tečku Ovaj nečin je mukotrpan i što je mnogo važnije ne deje dobre rezultete kade je potrebna animacija Često su oblici koje treba prikazivati relativno meli (figura čovaka u pakmenu, svemirski brod u Invajdersima' vočka kod Horacija'), pa možamo de ih svedemo na veličinu jednog karektere koji ćemo jednoslevnim PRINT prikezivatl na ekranu gde god je potrebno Karakteri se kod Spectruma (i kod većine drugih sličnih kompjutera) datinišu na matrici 8×8 kao što ćemo videti iz sledećeg jednostavnog primare

Pretpostevimo da želimo da detinišemo karekter u obliku figure čoveke sa slike Najpre smo ga kao što se vidi upisali u matricu kvedretiće teko de su neka polja crna e neka bale (česlo je lako uklopiti neku složeniji obtik u pritično malu matricu — za ovu umetnost biveju angažovani i profesionalci kade se pišu komercijelni progrem) Zatim ga pretvaremo u brojeva i to red po red (vlasnici drugih računara će ušlede sebi dosta napora eko provere da se to na njihovom kompjuteru ne radi kolonu pa

Svaki vlasnik Spectruma je sigurno nabavio neki komercijalni program za akcionu igru (neki ih imaju na desetine, a možda i na stotine) i, upotrebljavajući ga, pozavideo njegovom autoru: kako je moguće da je igra ovako dobro zamišljena, da je ekran ovako efektan, da su zvučni efekti ovako dobri . . . Dobru akcionu igru, zaista, nije lako sastaviti — ne toliko zbog toga što je teško napisati program koliko zato što nije jednostavno zamisliti dobar i originalan scenario koga se neko drugi već nije setio. Kada zamislite scenario, realizacija programa je stvar tehnike, koja, kao što rekosmo, nije teška, ali je lakša ako sami ne morate da je otkrivate od samog početka. To ja i smisao ove škole akcionih igara.



...nu, što je vrlo čest slučaj) pri čemu ...zacrnjeno' polje zamenjujemo jedinicom, ...a ono koje to nije — nulom. Konstruktori Spectruma su nas snabdeli funkcijom BIN ...ća oveko dobijene binarne brojeve pre...ara u dekadne. U našem primeru dobi...

...red 1= BIN 00001000
...red 2 BIN 00010100
...red 3 BIN 00001000
...red 4 BIN 00111110
...red 5 BIN 00001000
...red 6 BIN 00011100
...red 7 BIN 00100010
...red 8 BIN 00100010

Najbolje je da kada smo ispisali ove brojeva, upostimo računar na njihovom prevođenju u dekadne. Za prvi red otkucamo jednostavno PRINT BIN 00001000 i dobijemo broj 8, za drugi broj 20 i tako dalje. Zatim unosimo jednostavan program koji što je sledeći startujemo ga i, od tog momenta, karakter „A" postaje čovečuljak sa slike.

Da bismo vam olakšali pisanje sopstvenih akcionih igara dajemo tabelu sa nekoliko predloženih karaktera. Tu ćete naći sadržaje koje treba uneti u DATA linije he kreju programa da biste imali definisane oblike koji se često koriste. Jasno je naravno, da korišćenje karaktere koje će poznavati i hiljade drugih nije neka prevelika originalnost ali ćete lako promeniti neka detalje dobiti mnogo originalnije oblike. Kada se radi na maloj matrici (u ovom slučaju 8×8), čak i promena jednog polja iz temelje menja čitav utisak o karakteru. Dajte dakle na volju vašoj umetničkoj mašti!

Zvučni efekti

Nijedna komercijalna igra ne vredi mnogo bez interesantnih zvučnih efekata. Konstruktori Spectruma su na žalost ostavili jedino ne naročito fleksibilnu naredbu B... oja ima dve argumenta razdvojena

TABELA 1

```
IN63486   keys 1,2,3,4,5
IN64510   keys Q,W,E,R,T
IN65022   keys A,S,D,F,G
IN65278   keys Caps Shift,Z,X,C,V
IN61438   keys 6,7,8,9,0
IN57342   keys Y,U,I,O,P
IN49150   keys N,J,K,L,ENTER
IN32766   keys B,N,M,Symbol
          Shift,Break/Space
```

SLIKA 2

```
  10 GO SUB 1000
  15
  20 BEEP .1,10
  25
  30 IF m=1 THEN OVER 1
  40 PLOT INK ink,x,y: LET a=x
LET b=y
  50 LET x=x+(IN 65022=251 AND x
<254)-(IN 65022=254 AND x>1)
  60 LET y=y+(IN 64510=253 AND y
<166)-(IN 65278=251 AND y>1)
  70 PLOT INVERSE (m=2); INK ink
,a,b
  80 LET a$=INKEY$
  90 IF a$>="0" AND a$<="7" THEN
 LET ink=VAL a$
 100 IF a$="8" THEN PRINT AT 0,m
*11; FLASH 0; OVER 1; INVERSE 0;
"          ": LET m=m+1-(3 AND m
=2): PRINT AT 0,m*11; FLASH 1; O
VER 1; INVERSE 0;"          "
 110 OVER 0: GO TO 30

 120
1000 LET x=127: LET y=87: LET in
k=0: LET m=0
1010 BORDER 0: PAPER 7: INK 0: C
LS
1020 PRINT '"Kretanje """ W-gor
e","X-dole","A-levo","O-desno"
1030 PRINT '" + dijagonalne komb
inacije"
1035 PRINT '" R-Promena opcije"
1040 PRINT """"      "Pritisni ne
ki taster."
1050 IF INKEY$<>"" THEN GO TO 10
50
1060 IF INKEY$="" THEN GO TO 106
0
1070 CLS
1080 PRINT FLASH 1 "CRTANJE."; F
LASH 0;"  POMERANJE.  BRISANJE.
"
1090 PLOT 0,167: DRAW 255,0
1100 RETURN
```



Važnije od tehnike. Maštovitost i inventivnost predstavljaju ključ za dobru akcionu igru — sve ostalo se postiže gotovo samo po sebi

zarezima: prvi određuje trajanje tona, a drugi frekvenciju. Kao što vidimo, ne posoji mogućnost da se ton u toku izvođenja menja bilo po emplitudi ili po frekvenciji. No, to je tek sporedni problem sa naredbom BEEP; glavni je okolnost da računar ne može ništa da radi dok svira. Sa jedne strane, što je igre zanimljivija u njoj ima više zvučnih efekata ili je sa druge strane i manja dinamične jar računar gubi vreme na nešto što je, ipak, sporedno.

Kada pišemo bejzik program sa zvučnim efektima, najbolje je da koristimo READ i DATA naredbe koja nam omogućavaju da uz minimum mamorijskog utroška proizvedemo druge melodije. Na slici je dat jedan jednostaven primar zvučnog efekta koji bi mogli da otvaraju napadači iz vasione u toku spuštanja.

Vidimo da ze svaki ton moramo da navedemo dva člana DATA liste što nije samo veliki utrošak memorije već i gubitak vremene (READ nije ni izdaleka jedna od najbržih Spectrumovih naredbi). Zato je dobro odreći se promenijiva dužine tonova i nadoknaditi je „kompozitorskom veštinom", kao u drugom primeru.

Na kraju poglavlja o tonovima jedan praktičan savet koji je više hardverske prirode. U prodaji se nalaze tekozvene zvučne kutije za Spectruma („Sound boxse"). Njihava uloge je da pojačaju tonove koje računar generiše, čim zvučni efekti postaju interesantniji ili bar privlače pažnju šireg auditorijuma. Ovakve kutije možete sasvim uspešno zameniti baz likakvog posebnog troška. Povežite jednostavno MIC ili EAR na Spectrum sa MIC na kasetofonu, pritisnite REC i PLAY (bez umetnute kasete) i — to je sve. Zbog ekonomičnosti nekih rešenja Spectrumovog hardvera, kasetni ulez i izlez i generator tonova su tesno povezani, što omogućava da se kasetofon koristi kao pojačalo.

Crtač u računaru

Svi znamo kako u bioskopu vidimo „pokretne slike" — svake sekunde preko ekrana „proleće" 24 nepomične slike, ali mi zbog tromosti oka, imamo utisak kretanja. Jasno je da nešto slično možemo da pokušamo kod računara, ali jednostavni eksperimenti pokazuju da su šanse samo teorijske: računar, pre svega, ne može da u jednoj sekundi prepiše 10 (još manje 24) sedržaja ekrana koji inače zaprema 6 Kb RAM-a. Osim



SLIKA 4

```
9000 RESTORE 9100
9010 FOR n=0 TO 7
9020 READ a
9040 POKE USR "a"+n,a
9050 NEXT n
9070 RETURN
9100 DATA 8,20,8,62,8,28,34,34
```

SLIKA 5

```
RESTORE
FOR i=1 TO 7
BEEP duzine,nota
NEXT i
DATA .2,.7,.1,.2,.1,.2,.2,.4,.4
.2,.6,.2,.7
```

...ga mamorne od 48 Kb može de prmi... ...no šest ekrena što nije dovoljno ni za ...a sekunde televizije. Pe..keko je onde ...mpjuterska enimacije uopšte moguća? ...na je, naravno u tome što najveći deo ...rana osteje statičan a pokreću se samo ...delovi (ne primer, renije definisani ...akteri) i to na proizvoljno nago teko da ...jihov položaj u svekom trenutku može ...ti iz prethodnog položaje. Progre... ...treba de buda toliko vešt de promene boje istakne delove ekrena koje se ... a nautrelišе steličnu pozedinu. To ...more uvek de bude mnogo teško leko ...ko zahteve odredeno iekustvo. Mr ... počali od najjednostevnijeg karakte... ...koji se kreće po ekranu.

Posmatrejmo primer sa prve slike koji ...vlja najjednostavnju primenu ...ovne ideje. Ako otkucate progam vide... ...samo tečku koja naprijelno treperi ...? Naredba 30 se izvršave odmah posle ...edbe 20 što znači de kvedretić biva ...san prekličnо trenutno. Sa druge stre... ...bi kvedretić bilo ponovo ispisan ...ner treba de obavi jedno sabirenje i ...u GO TO naredbu što, naravno treje ... vreme. Tečka je dakle mnogo ...e uključena nago što je uključena i to ...zvodi naprijaten efeket.

Zadovoljstvo bez premca. Za mladog programere nema uzvišenijeg trenutke od momente kada na ekranu oživi njegove sopstvena igre.

Ako već na možemo da ubrzamo sabira...nje i GO TO naredbu, možemo bar da produžimo intervel koji tečka provodi na ekrenu dodevši jednu „mrtvu petlju". Stvar možemo de začinimo i se malo muzika.

```
24 FOR N=1 TO 5    NEXT N
27 BEEP .01 RND 20
```

Sade je već mnogo bolja el se ispojava novi problem čim komjuter pokuše da štempe karakter izvan ekrana (ne to ne treba tako dugo čekati). biva prijevljena greške. Prve ideja koju bramo mogir da primenimo je IF naredba koje ispituje de li se stiglo do kreje ekrene. No Spectrumova IF naredba je spore i bolje br bilo iskoristiti FOR-NEXT petlju (9 bi išlo od 1 do 31). etl postoji i elegantnrje rešenje de karakter koji se kreće kede doda do jednog kraje ekrana, neslena i pojevi se na suprotnom. Ne sledećoj slici je dat progrem koji to omogućave. Početna linije izgledaju sasvim jesno sve dok ne stignemo do linije 40. Temo nes očekuje neobične konstrukcije. LET x=x+(y=31). Šte je to?

Malo bolji pozneveoci Spectruma znaju de se l logički izrezi izražavaju brojevima, (y=1) je logički izrez koji može de bude tačan ili lažen. Ako je tačan računer će mu dodeliti broj 1 e eko je netačan — broj nule (kod mnogih drugih računere situacije je potpuno iste osim što se tečnom iskazu dodeljuje vrednost −1. ovo ćete najlekše ispiteli ako otkucate PRINT (3=3) r pogledate rezultet). Kade dakle y postana 31. x će biti povećano ze jeden što znači de će računar početi de piše od početka sledeg rede.

Drugi deo naredba 40 izgleda još zagonatnrje jer je ponovo iskorišćena naredba AND. Smisao ove naredba je de najpre lo rešćrstimo da računer na prijevi grešku kada tačka, liniju po liniju, etigne do kreja ekrena. U tom slučaju bi trebalo de x ponovo postane nule tj. da se tečka pojevi u gornjem ugtu ekrena. Da vidimo šte se dešave u oba kritične slučaje kade je x jednako 22 i kade nije.

Ako je x jednako 22 onde l x=22 (obrelite pažnju na redosled operecije) ima vrednost 1 koje, logički pomnožena sa 22 deje opet 22 što smenjuje x ne nutu. Ako pak x nije 22 X=22 će dobiti vrednost nule pa će nule logički pomnožena se 22 dati ponovo nulu koja neće menjati vrednost X. Ako ste shvetili ovu logiku, ne bi trebalo de imete problema sa razumevanjam linije 50 koje provereva da li je stiglo do kreje ekrene. Tu liniju ćemo vam, dakle, ostevili za vežbu.

Glavom o zid

Teko smo na reletivno zadovoljavajući način rešili problem kretenja tačke po jednoj liniji. Jasno ja, međutim, da nam to nije dovoljno — potrebna nam ja tačka koja će se mnogo slobodnje kreteti po ekranu i odbijeti se od drugih objekata ne njemu. Za to moremo uvesti promenljivu koje će govoriti o pravcu kretanje. Za slučaj da se kretanje obavlje po jednoj prevoj, ta promenljiva će se posle svekog štempenja i brisanja dodati trenutnoj vrednosti iksa (eko promenljiva ima vrednost 1, karakter će se kretati po jedno polje udesno), dok se promena smere postiže jednostavnom promenom znaka promenljive. Ovej slučaj je, međutim, previše jednostavan pa ćemo se pozabaviti kretanjem u evim smerovima. Njege karekteriše promenljiva d, koja će imeti vrednost 1 za kretanje dole, 3 ze gore, 0 za desno i 2 za kretanje levo (za trenutek ćemo videli zašlo su brojevi teko rasporedeni). a x i y će kao i do sada označavali položej tečke na ekrenu.

```
1000 LET X=X+(d=1 OR d=2)−(d=0 OR d=3)
1010LET y=y+(d=2)−(d=1)
```

Ukoliko karekter „putuje" preko ekrena i udari u horizontalni zid d treba da promeni vrednoet. To na primer možemo de postignamo naredbom

```
2000 LET d=d+1−(2 AND (d=3 OR d=1))
```

Ako ste početnik ova naredba mogu de vas zaptaše. No na brmite nje teško razumstr njihvo dejstvo dovoljno je de se odlučite za naka konkretna vrednosti x y i d i ručno korak po korek izračunate kako će se one promeniti posle izvršavanje ovakvih naredbi. Posle najviše dva prmere sve će biti sasvim jasno!

Krajnja je vreme de izložimo nekoliko nečine ne koje može de se utvrdi de li je došlo do sudare dva objekte koje se kreću po ekrenu. U nakim slučajevima možemo jednostevnim IF uporeditr promenljive u koje su upisane njihove pozicije. U mnogim slučajevima, međutim, nije ekonomično trošrti memorijski prostor na koordinate svakog objekte (zamislimo samo de pišemo

program u koma igrač treba da proda kroz prebukrreni pojas asteroide) kede je on već upisan ne ekran odekle ge uz malo veštine možemo pročitati

Prvi način da to učinimo je naredba SCREEN (x, y) gde su x i y koordinate pozicije na ekranu koju želimo da pročitemo Naredba će vratiti karakter koji se nalazi ne tom mestu uz dve ogrenrčenja koje su veoma ozbiljna: predefinisani i mvarzni karakteri ee nekorektno detektuju

Drugi nečin je naredbe ATTR koje isti-na ne deje obaveštenje o karakteru na nekom mestu air zato govori sva o njegovoj boji Ukoliko uspemo da svaku vratu obje-kata obojimo na odredani način ATTR će nam pomoći da se orijentišemo Kao i SCREEN ATTR ma dve argumente koji odreduju mesto na ekranu Kao rezultat se dobija jedan bajt spakovan keo zbir sledećih sabrreke

128 eko karaktar blinkuje
64 ako je kerakter posebno sveteo
8 puta boja pepire (PAPER)
1 puta boja mastila" (INK) kojim je karakter pisan

ATTR na obraća pažnju na to da li je karekter stenderden lil ge je korisnik sam definisao iz jednostevnog razloga što i ne enelizire tej kerekter — keo što joj i ime govori ATTR enalizira etribute karektere (o etributma smo dosle govorili u „Računerime 1 kade smo obješnjevali kako Spectrum kodrre sadržej ekrena)

Poslednji način predstevlje naredba POINT koja nije u velikoj upotrebi teko je neobično moćna Prete je dve argumente koje deju poziciju na ekranu ali ne pozicliju karaktera nego pozicrju tečke kao kod neredba PLOT x i y POINT vraća broj 1 lili0 — jedan ako je tačka čije su koordinate date osvetljena, e nule eko je zetemnjene Zašto je ove močna neredbe kojom možemo de ispitemo status bilo koje tačke ne ekrenu manje u upotrebi? Jednostevno zato što su bejzik programi sporr i ne mogu sebi de dozvole luksuz rede sa tečkama — redi se isključivo sa karekterima Primenom POINT možemo jesno, de ispitemo svaki karekter tečku po tečku, ell tekvo ispiteivenje oduzima previše vremane de bismo mogli da ga preporučimo

Da ilustrujemo sve ovo objevijujemo program Lopte i reket' Obzirom de se kreću samo dve objekte (lopte i reket) igre je prilično brza i dinamične Kao i obično na treba samo da sa zabevljate njime (za zebavu ima i mnogo boljih igara) nego da pokušate da ga analizirate u čemu će nedemo sa pomoći i komentari uključeni u REM linije

Skrolovanje

Skrolovanje je trenutak kada sa evi objekti (slova, specrjalni karakteri i crteži) pomeraju nagora za jedan red da bi se oslobodio prostor za novu liniju Za normelen rad i pisanje programa (posebno kod kompjutera kod kojih se neredbe kuceju po čitevom ekranu kod Spectruma su za naredbe odvojeni samo poslednji redovi) skrolovenje je evekako neophodno Spectrum medutim posle ispisivenje svake 22 linije očekuja od korienika de potvrdno odgovori na pitanje Scroll? (ze potvrdan odgovor se smatra pritisak na bilo koji



Iz jedne avanture u drugu. Prizori iz igračkog života

```
SLIKA 6

2000 FOR n=10 TO -10 STEP -.5
2010 BEEP .02,n
2020 NEXT n

SLIKA 7

  10 LET y=0
  20 PRINT AT 10,y,"."
  25 PAUSE 1
  30 PRINT AT 10,y," "
  40 LET y=y+1
  45 IF y>31 THEN  GO TO 10
  50 GO TO 20

SLIKA 8

  10 LET x=0  LET y=0
  20 LET e=x' LET b=y
  30 PRINT AT x,y," "
  40 LET x=x+(y=31)  LET x=x-(22
AND x=22)
  50 LET y=y+1-(32 AND y=31)
  60 PRINT AT a,b," "
  70 GO TO 20
```

tster osim na N' i BREAK) što može da bude vrlo neprijetno za ekcione rgre Jedan od načine de izbegnemo ovu sistemateku poruku je da na samom početku programa upotrebimo POKE 23692, −1

Skrolovanje je vrlo pogodno ako se neki objekt kreće prema vrhu ekrena uz umereno pokretanje po s oei Jedan od najjednostavnijih načine de isprobate skrolovenje je da štampate upravo definisani karektar u obliku melog svamirskog broda u posteđnjem redu e zatim koristite prezne' PRINT neredbe koje će ge pomeriti prema vrhu ekrena Videćete de je pokretanje broda daleko brže nego u renijim primerime gde emo koristil PRINT AT Razlog je u tome što skrolovenje obavlje mašinski programm koir je mnogo brži od svakog bejzike

Kade smo se uverili u korisnost skrolovenje poželećemo de ga upotrebimo i po x dimenzijr (sleva na desno lil sdesna nelevo) Ne želost mašinski polprogremi koji bi ovo omogućili ne nelaze se u ROM-u e sveki bejzik koji treba da pomeri 6 K video memorije bi bio toliko spor da ga ne treba ni pominjeti Ipak kao vežbu pokušajte de nepišete bejzik program koji pomere ekren ze jedno mesto ulevo Ako uspete zesluživate čestitke i kompliment de je krejnje vreme de definitivno prađete ne mašinec

Ako nemate volje ze pisenje mašinskih programe dejamo vam dva koje će vam lepo poslužiti prvi omogućava levi a drugi desni skrol Kao što se vidi ova dve progrema ukupno imaju samo 46 bajtova što znači de eu toliko optimizovenr de se početnici teško mogu upustiti u njihovu analizu zato i ne dajemo disasembtirani listing

Glevna nevolja sa skrolovanjem je što se ne pokreće samo ono što bismo mi želeli Na ekranu se nalma obično nalazi i neki broj bodove meksimelni skor ebatreki uputstreve za upotrebu ili nešto slično Sukcesivno pomerenje ekrena izeziva jesno i gubljenje ovoga teksta što je prilično neprijetno U komercijelnim igrema se često koriste mašinski potprogremi koji skrolu-ju samo deo ekrana ell se u najvećem broju slučajeva može proći i bez njih jednostavno koncipirejte sadržaj akrane tako da su informacije koje stelno treba prikazivati u prvom ili eventualno prve dva rede e zatim ih ponovo štampajte svaki put kade iskoristite PRINT da bi ekran skroloveo Na taj način ujedno rešavate i problem promene skore u toku igre a Spectrum je dovoljno brz da se ovo stelno ispisavanje mnogo ne primeti

Strogo kontrolisani bajtovi

Izgleda de smo ee približili mešinskom jeziku — došli smo do naredbi kojima možemo de pristupemo čitavoj memorij naseg Spectruma Ona su neobično močna ell imaju i loša strane dok sa do sada operetivni eistem našeg računara brinuo o svim greškema i čuveo naš program PEEK i posebno POKE su naredbe koje mogu da budu destruktivne Zato svekl program ih koristi treba pra ateriovenje svekaa snimiti na traku — to je mere predostrožnosti koja nikade nije suvišne

Nećemo govorit o sintaksi i funkcijama ovih naredbi (obješnjena su na sa početku Mallh tajnt velikth mejstore programirenja' u „Računarima 1) ali mo mo da kažemo nakoliko reči o specifičnostima Spectrumova memorijske mape Treba nelma, da nademo memorijake proc u koje možemo slobodno de smeštemo sadržaj ne plešeći sa da će naš program biti oštećen U nekim programima ćete naići korišćenje REM linija u tu avrhu što direktan znek de su autori tth programa pre Spectruma imati ZX81 lako vam bismo mnogo preporučili ovaj način ćemo nekoliko reči o njemu pre nago predemo na naredbu CLEAR

Obrišite program koji se nalazi u r

naru i otkucajte prvu naredbu i REM AAA a zatim POKE 23760, 127. Listanje programa će otkriti da se prvo slovo A pretvorilo u znak za kopiranje? Pogled na tabelu ASCII karaktera otkriva da je 127 kod tog znaka što znači da smo pronašli mesto na koje je smešteno slovo A i promenili ga pomoću POKE! Adresa 23760 odgovara prvom karakteru iza REM naredbe (podrazumeva se da ispred REM-a nema drugih naredbi). Ne moramo da se ograničimo na jedan karakter prve REM linija može da ima mnogo slova i tako poslužiti kao akumulator za čitav mašinski program ili labelu podataka.

Zašto smo rekli da ne preporučujemo ovaj metod? Kada jednom počnete da listate program prve linije će biti puna svakakvih karaktera. Rešenje ovog estetskog problema bi moglo da bude smeštanje mašinca u poslednje linije programa ali je njih mnogo teže pronaći u memoriji računara i što je još gore njihove početna adresa se menja svaki put kada se program produži ili skrati.

Najjednostavniji način da rezervišete prostor u memoriji je naredba CLEAR. Za Spectrum 16 Kb CLEAR n će rezervisati prostor od adrese (n+1) do 32599 za mašinski program i podatke ukoliko je n veće od 32600 neki ili svi karakteri koje ste definisali mogu da budu izgubljeni. Sve ovo se odnosi i na Spectrum 48 Kb. s tim što su sve cifre veće za 32768. U memorijski prostor rezervisan na ovaj način ćete za sada smeštati samo sadržaje delove ekrana koje treba brzo i dinamično razmenjivati da bi se dobila animacija, ali ćete ubrzo naučiti da je mnogo rentabilnije izbeći korišćenje matrica i nizova kada su svi njihovi elmenti između 0 i 255. Takve brojeve je bolje smeštati u rezervisani deo memorije. pri čemu svaki od njih zauzima po jedan bajt. Vreme utrošeno za pisanje programa koji će po potrebi preračunavati poziciju nekog bajta i pozivati ga će se višestruko isplatiti — najpre zbog utroška memorije a zatim što je još važnije zbog mnogo brže izvršavanja programa.

Razbijanje nizova na bajtove je samo jedna (i to marginalna) upotrebe naredbi PEEK i POKE. Mnogo je zanimljivije koristiti ih za promenu sadržaja sistemskih promenljivih. Govorićemo samo o onima koje su od posebnog značaja za akcione igre.

Već smo videli da smeštanje broja različitog od 1 u memorijsku ćeliju 23692 sprečava računar da ispisuje Scroll? kada se ekran popuni. Slično lome POKE 23609,100 (probajte i druge brojeve umesto "100") daje kratak ton po pritisku svakog tastera umesto uobičajenog klika što je zgodno u toku INPUT-a u neku igri.

Promena sadržaja memorijske ćelije 23624 utiče na boju kojom se poruke ispisuju na ekranu. Ova memorijska ćelija čuva čuve atribut alove ovih poruka na način koji smo upoznali kod naredbe ATTR. Slično lome memorijska ćelija 23693 je odgovorna" za sadržaj viših redove ekrana atribute tog sadržaja. ali sa kojima dosta teže pokušaje na primer. da izvršite POKE 23659,1 e zatim i CLS.

Ćelije 23677 i 23679 su korisne za one koji koriste naredbu DRAW koja je keo što znamo, „relativna — pomeranje se obavlja u zavisnosti od položaja poslednje pome-

```
  9 REM Inicijalizacija
 10 GO SUB 5000
 19 REM Stampanje lopte,cuvanje
       starih vrednosti
 20 PLOT x,y  LET x=x  LET by
 30 LET x=x+dx  LET y=y+dy
 39 REM Provera polozaja lopte
 40 IF y<6 THEN GO SUB 2000
 50 IF x<6 OR x>240 THEN GO SUB
3000
 60 IF y>h THEN GO SUB 4000
 69 REM Pomeranje reketa
 70 PLOT p,6  DRAW 20,0
 80 LET p=p+(5 AND INKEY$="6" A
ND p<235)-(5 AND INKEY$="5" AND
 p>0)
 90 PLOT p,6  DRAW 20,0
 99 REM Obrisi loptu
100 PLOT a,b
110 IF true THEN GO TO 20
119 REM Lopta promasila reket
120 LET bali=bali-1  LET s=s-1

130 PRINT AT 0 10-LEN STR$ s;  O
VER 0 s
139 REM Da li je kraj?
140 BEEP .1 0:  IF bali=0 THEN GO
 TO 9000
199 REM Priprema za sledecu
       loptu
200 PLOT p,6  DRAW 20,0
210 GO SUB 5300
220 GO TO 20
1999 REM Pomodjen reket
2000 LET dx=RND*SGN dx*4+SGN dx*
2
2010 LET dy=AND*4+2
2020 LET true=(x>=p AND x<p+21)
2050 LET y=7
2070 LET s=s+1:  PRINT AT 0,10-LE
N STR$ s;  OVER 0,0
2100 BEEP .01 20
2200 RETURN
2999 REM Pomodjen zid
3000 LET dx=-(RND*SGN dx*4+SGN d
x*2
3010 LET dy=RND*SGN dy*4+SGN dy*
2
3020 LET x=6+(242 AND x>240)
3100 BEEP .01,0
3200 RETURN
3999 REM Pomodjen vrh
4000 LET dy=RND*-4-2
4010 LET dx=RND*SGN dx*4+SGN dx*
2
4020 PLOT 6,h  DRAW 243,0
4030 LET h=h-(2 AND h>30)
4040 PLOT 6,h  DRAW 243,0
4050 LET y=h+1
4100 BEEP .01 40
4200 RETURN
4999 REM Potprogram za
       inicijalizaciju
5000 LET bali=3  LET s=0
5010 LET h=166
5100 BORDER 4  PAPER 1  INK 7  C
LS

5300 LET x=127  LET y=h-1
5310 LET dx=RND*10-5  LET dy=-(6
ND*4+2)
5320 LET p=115
5330 LET true=1
5600 GO SUB 6000
5700 RETURN
5999 REM Crtanje ekrana
6000 OVER 0
6010 PRINT AT 0,1,"SKOR=000
 OSTALO LOPTI 000"
6020 PRINT AT 0,10-LEN STR$ s;s
 PRINT AT 0 31-bali, FLASH 1;"O"
6100 PLOT 5 0  DRAW 0,167  DRAW
245,0  DRAW 0,-167
6150 PLOT p,6  DRAW 20,0
6200 OVER 1
6400 RETURN
7999 REM      *** KRAJ IGRE ***
8000 PRINT AT 10,11, FLASH 1,"KR
AJ IGRE"
8010 BEEP 1,-12  BEEP 2,-24

8020 OVER 0  BORDER 7  INK 9
8030 STOP
```

```
  5 REM Prostor za "masinac"
 10 CLEAR 31999
 11 REM 64999 umesto 31999
       za 48K
 20 GO SUB 8000  REM "masinac"
8000 RESTORE 8100:  FOR i=32000 T
O 32045:  READ a:  POKE i,a:  NEXT
 i:  RETURN
8009
8010 REM            SCROLL levo
8019 REM RANDOMIZE USR 32000-
8020 REM 65000 umesto 32000
       za 48K
8040 REM            SCROLL desno
8049 REM RANDOMIZE USR 32023-
8050 REM 65023 umesto 32023
       za 48K
8100 DATA 1,255,2,17,0,88,33,1,0
8,237,176,1,0,24,17,0,64,33,1,64
,237,176,201
8110 DATA 1 255,2,17,255,90,33,2
54,90,237,184,1,0,24,17,255,87,3
3,254,87,237,184,201


SLIKA 11


  1 REM Za 16K SPECTRUM
  2 CLEAR 32592  REM
       (65300 za 48K)
  3 LET num chars=1
  4 LET bytes reserve=num chars
*8
  5 LET n=32599-bytes reserve
    REM  (65367 za 48K)
  6 LET char before first=CODE
"#"-32
  7 LET byte before first=char
before first*8
  8 LET char=n+1-byte before fi
rst-256
  9 LET chr in 23606=char-256*I
NT (char/256)
 10 LET chr in 23607=INT (char/
256)
 20 FOR i=n+1 TO n+bytes reserv
e:  READ a:  POKE i,a:  NEXT i
 30 DATA 255,36,36,24,24,36,36,
255,0

 50 POKE 23606,chr in 23606
 60 POKE 23607,chr in 23607
 70 PRINT AT 11,15,"#"
 80 POKE 23606,0
 90 POKE 23607,60
```

nute" tačke. Ako, na primer, najpre izvršimo PLOT 100,80 a zatim DRAW 20,60 linija koju dobijemo se neće završavati u tački 20,60 nego u 120,140! Promenom sistemskih promenljivih 23677 i 23678 možemo da simuliramo apsolutni DRAW. U prethodnom primeru bilo bi dovoljno zameniti naredbu DRAW 20,60 naredbom DRAW 120-PEEK 23677, 140-PEEK 23678 ili, alternativno zamenimo ovo sa dve naredbe od kojih bi jedna izračunavala a druga crtala. Što je način koji će početnici uvek radije primeniti.

Peščani sat

U mnogim igrama je potrebno meriti vreme koje je igraču bilo potrebno da obavi neki zadatak. Poneko to radi tako što u pravilnim intervalima, dodaje 1 nekoj promenljivoj, ali se ubrzo pokazuje da je ovakav set više nego netačan, što izaziva na malo zaprepašćenje. U čemu je stvar? Vaš Spectrum ne radi neprekidno posao koji ste mu vi zadali, vrlo često (50 puta u sekundi) on

SLIKA 12

```
  1A REM *** RELI ***
9998 REM Definisanje karaktera
9999
9000 CLEAR 32399  RESTORE 9200
9020 PRINT AT 11 10  FLASH 1 "SA
CEKAJ"
9030 FOR i=32400 TO 32599
9040 READ a  POKE i a  NEXT i
9050 FOR i=1 TO 10  FOR n=0 TO 7
9060 READ a  POKE USR CHR$ (i+14
9)+n a  NEXT n  NEXT i
9070 CLS   PRINT AT 9 7 "STARTUJ
 TRAKU"
9090 LOAD ""
9100 REM Znaci za definisanje
     karaktera
9200 DATA 0 24 36,36 126 102,102
,102
9210 DATA 0 124,68,68,126,70,70,
126
9220 DATA 0 124 64 64 96 96,96 1
26

9230 DATA 0 126 66 66 90 90,90 1
26
9240 DATA 0 126 64,64,120,96,96,
126
9250 DATA 255,129,189,165,165 18
9 129,255
9260 DATA 0 126 64 64 96,102 98
126
9270 DATA 0 68 68,68,124 76,76,7
6
9280 DATA 0,16,16,16 24 24,24,24
9290 DATA 204 204,51 51 204,204,
51,51
9300 DATA 0,0 0,0,0 0,0,0
9310 DATA 0 32 32,32 48 48 48 62
9320 DATA 0,126,66 66 66 66,66 6
6
9330 DATA 0,124,76,76,76,76,76,7
6
9340 DATA 0,126,66,68,98 98,98 1
26
9350 DATA 0,124,68 68,124 64,64,
64

9360 DATA 0,0,0,0,0,0,0,0
9370 DATA 0 124,66,68,126,70,70,
70
9380 DATA 0,124,64,64,124,12,12,
124
9390 DATA 0,126,16,16,24 24 24,2
4
9400 DATA 0 68 68,68,68 68,68 12
4
9410 DATA 0 126,66,2 30 16 0,16
9420 DATA 0 66 66,66 66 66,66 12
6
9430 DATA 0,24 24,0,0,24,24,0
9440 DATA 0,68,68,124,16,16,16,1
6
9499 REM Podaci za "kola"
9500 DATA 0,73,73,93,127,93,65,6
5
9510 DATA 33-66,188-60,60,29,2,4
9520 DATA 254,16,56,63,56,16,254
,0
9530 DATA 4 2,29,60,60,188 66,33

9540 DATA 65,65 93,127-93 73,73,
0
9550 DATA 32-64,184 60 60 61,60,
132
9560 DATA 127,8,28,252,28,8,127,
0
9570 DATA 132,66,61,60,60,184,64
,32
9599 REM Podaci za broleve
9600 DATA 0 60 66,66,90 90,90 60
9610 DATA 0 0 0 0,12 12 12,12
9620 DATA 0 124 2,2 60 46 56 62
9630 DATA 0,124,2,2 60,6,6,126
9640 DATA 0,64 96,108,100,126,4,
4
9650 DATA 0,62,64,64,60,6,6,126
9660 DATA 0,62,64,64,124,70,70 1
26
9670 DATA 0 126,0,8 12 12,12,12
9680 DATA 0 60 66,66 60 70,70 60
9690 DATA 0 60 66 66 60 6,6,126
9999 POKE 23606,0  POKE 23607-60
  RETURN
```

BLIKR 13

```
  5 REM **** RELI II ****
  10 GO TO 300R
  100 LET a1=IN 64S1A-255  LET d1
=IN 65822-255  LET mv=IN 57342
  120 LET dr=dr+(mv=254)-(mv=253)
-(6 AND dr=7 AND mv=254)+(8 AND
dr=8 AND mv=253)
  140 LET x=x-(dr=0 DA dr=1 DA dr
=7)+(dr>2 AND dr<6)  LET y=y-(dr
>4 AND dr<=7)+(dr<4)
  160 PRINT AT x,y CHR$ (144+dr)
  180 IF a1 OR d1 THEN PLOT 4+a*8
,56  DRAW 0 5  LET a=a-(a1 RND a
(4)+(d1 RND a>1)  PLOT 4+a*8 56
 DRAW A,5
  200 FOR i=1 TO 5-a  BEEP .02,RN
D*20-40  NEXT i
  220 PRINT AT x y,CHR$ (144+dr)
  240 GO TO LOOP+(400 AND ATTR (x
 y)<>87)
  500 GO TO END
  997

  998>REM Alternative
  999
1000 POKE 23606,136  POKE 23607
123  RETURN
1497
1498 REM Normalno
1499
1500 POKE 23606,0  POKE 23607 60
  RETURN
1997
1998 REM Stampanje vremena
1999
2000 GO SUB ALTER
2010 LET s$=c*"0"  RND m(10)+STR$
 m+c*"0" AND s<(10)+STR$ s
2020 PRINT RT x,b,  INK 3,CHR$ (C
ODE s$(1)+104) CHR$ (CODE s$(2)+
104)
2030 PRINT AT a b+3 CHR$ (CODE s
$(3)+104),CHR$ (CODE s$(4)+104)
2040 GO SUB NORM
2050 RETURN

4997
4998 REM ***INICIJALIZACIJA***
4999
5000 LET h=1E30
5010 LET LOOP=100
5020 LET ALTER=1000  LET NORM=15
00
5030 LET TIME=2000
5040 LET START=5300  LET NEXT CA
R=5400
5050 LET NEW GAME=9000  LET BYE=
9900
5060 LET END=7000
5070 LET CRASH=7500  LET FINISH=
6000
5100 POKE 23693,33  POKE 23659,1
' CLS  INPUT '
5110 GO SUB ALTER
5120 DIM b$(2,5)  LET b$(1)=CHR$
 16+CHR$ 2+CHR$ 17+CHR$ 64+CHR$ 1
02' LET b$(2)=CHR$ 16+CHR$ 1+CHR
$ 17+CHR$ 7+CHR$ 107

5130 BRIGHT 0  OVER 0  FLASH 0
INVERSE 0  BORDER 5  PAPER 7  IN
K 1
5140 PRINT AT 0,0'  RESTORE 5500
  LET n=1
5150 FOR i=1 TO 189  READ a  LET
 n=n-n+NOT n  FOR v=1 TO a: PRIN
T b$(n+1)'  NEXT v  NEXT i
5160 PRINT RT 19,10  INK 0,"J",
AT 20 10 "J" AT 1,2, INK 3,"PRV
Okvvvvv" AT 6 6 "VREME",AT 1,14,
 INK 1 "SPECTRUMkSPEEDWAY",AT 13
,1 "gear"
6170 FOR i=1 TO 4  PLOT 4+i*8 62
: NEXT i  INK 6
5300 GO SUB ALTER  PAINT AT 8,6,
 INK 3 "IIxII"
5310 LET c=3  FOR i=1 TO c  PRIN
T RT 16+1,1 "C"  NEXT i  PLOT 8,
69  DRAW 31-0
5320 PRINT #0  INK 2, PAPER 6,"f
fffffffSPECTRUMkSPEEDWAYffffffff"

5330 LET a=0  LET s=0
5399 REM Sledeca kola
5400 LET x=20  LET y=10  LET dr=
6  LET s=1
5410 PLOT 12,56  DRAW 0,5
5420 OVER 1  GO SUB NORM
5430 POKE 23672,0  POKE 23673,0
 BEEP 1 0
5490 GO TO LOOP
6499 REM Podaci za ekran
6500 DATA 34,10,2 17 36,11,1,5,3
 4 7,12
6510 DATA 1 2,2 5,1,2,5,4,7,2 1,
2 8,2 6,3,2 5,1 2,1
6520 DATA 2,2,5,2,2 5,2,9 2,1,2,
1 3,1,2 2,3 4 2,3,8
6530 DATA 1 2 1,2,1,3,1,2 3,2,4,
4 7 2 1 2 1 3 1,3,2,2,4
6540 DATA 8 3 7 2 1 2 2,2,2 3 2,
4 8 1
6550 DATA 2 9 2 1,3 1,3 2 2 8,4
1 2 3 2,1,3,2 2

6560 DATA 2,2,3,4,3,2,1,3,1,3,2,
2,2,2,2,3
6570 DATA 4,2,3,1,6,2,3,2,2,2 3,
7 3 3 4,2,4
6580 DATA 1 3 2 3,6 3,9 4 2 3 3,
2 2 1 2 3 3
6590 DATA 3 3,3 4,2,2,1 2,3,7,4
3 3 5 2,2,1 2,13,4 5,1,4 6,11 6,
8 34
6997
7000 PLOT 4+a*8 56  DRAW 0,5
7010 OVER 0  LET s=s+INT ((PEEK
23673*256+PEEK 23672)/50)
7020 LET m=s+INT (s/60)  LET s=s
-m*60  LET a=8  LET b=5  GO SUB
TIME
7030 GO TO CRASH+(FINISH=CRASH A
ND KTTR (x,y>=56)
7497
7498 REM ********CRASH*********
7499
7500 FOR i=A TO 25  LET dr=dr+1-
(8 AND dr=7)  PRINT AT x y;CHR$
(144+dr)  BEEP .32 1  NEXT i

7510 GO SUB ALTER  PRINT AT x,y;
b$(1)  GO SUB NORM
7520 PRINT AT 16+c,1 " "  LET c=
c-1
7530 IF c>0 THEN GO TO NEXT CAR
7540 GO TO NEW GAME
7997
7998 REM ********FINISH*********
7999
8000 PRINT AT x y CHR$ (144+dr)
 LET t=60*m+s
8010 GO SUB ALTER
8020 INPUT '
8030 PRINT AT 1 0 #0 'jjjjjIkBRR
VO -USPELI STEKjjjjjjj"  PAUSE 2
00
8040 INPUT '
8050 IF t<h THEN LET h=t  LET a=
1  LET b=7: GO SUB TIME  GO SUB
ALTER  PRINT AT 1 0 #0,"ffffKI T
O JE NAJBOLJE VREMEkffff'  GO TO
 8070

8060>PRINT RT 1,0,#0 'RL1 VREME
NIJE NAJBOLJE"
8070 PAUSE 200
8080 PRINT AT x y,"J"
8090 GO SUB NORM
8097
8098 REM *****NOVA IGRA*****
8099
9065 INPUT "JOS JEDNOM (D/N)", L
INE z$
9010 IF (z$+" "(1)="N" THEN GO
TO 8YE
9020 INPUT "pr!ti=n! ENTER "  LI
NF z$
9030 GO TO STRAT
9500 BORDER 1  PAPER 1  INK 7  C
LS
9550 PRINT RT 10,13  FLASH 1  IN
K 6  PAPER 2,"ZDRAVOO"
9999 STOP
```



dobije takozvani interapt koji mu nalaže da prede ne izvršavanje jednog programa u ROM-u koji će pre nego što računar nastavi sa radom obaviti neku vrstu generalnog spremanja Bilo kakav vremenska petlja koju stavite u program će dakle biti mesto prekidana i taj broj prekida neće moći lako da izračunate

Zato je za merenje vremene najbolje koristiti sistemski časovnik koji računar povećava za jeden u toku svakog interapta To je jedna od komponenti pomenutog generalnog spremanja) Za upotrebu ovog časovnika treba upoznati memorijske ćelije čije su adrese 23674, 23673 i 23672 Svake sekundu sekunde sadržaj ćelije 23672 biva uvećan za jedan Kada se on poveće ne 255 računar povećava sadržaj ćelije 23673 dok se nula smešta u 23672 Slična operacija se ponavlja i kada sadržaj 23673 postane 255 samo što sa tade sadržaj 23674 povećave za jeden Sasvim je prirodno da vreme prosto izračunato vreme u sekundeme i kada su sva tri ova brojače bila nula (možete ih sami postaviti na nulu pomoću POKE na početku igra) LET sekundi = (65536* PEEK 23674+256 *PEEK 23673+PEEK 23672)/50

Na žalost Spectrumov sat je daleko od savršenstve Pre svega vreme biva izgubljeno svaki put kada se izvrši BEEP (do sade je ponovno inicijalizovenje nekih sistemskih promenljivih posle BEEP bilo od koristi), kada se koristi neka periferijska jedinica (npr štampač) Pored toga ponekad izračunato vreme može da bude netačno moguća sa sadržaj 23674 nije bio promenio dok je računar izvršavao prvo PEEK Zeto se u trenutku povećan e sadržaj 23673 anulira pre nego što je izvršeno drugo PEEK Zato se u odnosu sa upotrebu savetuje da se uvek računaju dva broje sekundi pa da se veći kao zatačan

Tenk na relju

Jeste li znali da možete da redefinišete više od 21 karaktera (čak do 96) i tako povećale raznovrsnost vaše ilote napadača iz svemira? Evo kako se to radi na najlakši mogući način

Najpre iscrtajte sve nove karaktere i izračunate dekadne vrednosti njihovih linija

— Odlučite koji originalni karakteri treba da budu predefinisani i na koji način

— Upotrebite CLEAR da rezervišete prostor za definicija Za svaki karakter koji treba definisati rezervišite osam bajte

— Prebrojte koliko ima karaktera pre onoga koji treba redefinisati (ako ne primer redefinišete znak "4" koji ima kod 42 neći ćete da pre njega ima 42—32=10 karaktera) Zatim pomnožite taj broj sa 8 i nazovite rezultat m

izvršite naredba

POKE 23606 m-256 INT (m/256)

POKE 23607 INT (m/256)

— Okončajte definiciju karaktera na uobičajeni način

Sva će ovo postati mnogo jasnije kada pogledate primar koji predefiniše karaktei u grafični blok

Golovo sva što smo do sada prešli sadržano je u igri Reli Treba naravno voziti automobil po kružnoj stazi Automobil doduše više liči na tenk eli je to zato što je potrebno da se ne neki način vidi kuda se krećemo iz oblika pokretnog predmeta tenk je za to neobično pogodan zbog „cevi Ukoliko vam se ovej oblik ne svide, ili ste mnioljubivo raspoloženi možete da ga promenite bez rkakvih problema

Piogrem koristi IN (naredbu opisanu na početku teksta) da omogući pritisak dva tastere u isto vreme ne primei tastere za ubrzanje i promenu pravca Automobil nairne ima četiri brzine Pritisak na O vas ubacuje u sledeću, e pritisak ne A u prethodnu brzinu Automobil se kreće u ovim prevcima tastei P je namenjen skretanju u nrevcu kazaijke na časovniku a O skretanju u obrnutom pravcu

U igri je korišćeno merenje vremena eli se vieme prikazuje tek kade nastiedate iir kade obidete čitav krug — Igra bi inače bila prespora Za ispunjenje zadetka imate tri automobila svaki put kada stiedete, računai ves vieća ne početek, što je dodelno oležanje

Glavne karakteristika Igre je predefinisani set karektera, praktično su eve slova i brojevi predetinisani uz korišćenje ideja iz piethodnog poglavlja To je ne žalost učinilo program piedugačkim pa je moreo de bude podeljen u dve dele prvi definiše karaktere e zatim učitave drugi koji predstavlja pravu igru Ovakva podela programa jesno ne odgovara viasnicime Spectruma 48 Kb koji mogu uz málo veštine ujediniti programe u jednu celinu

Ukoliko kucata program iz dva dele otkucajte najpre pivi snimite ga sa SAVE „Reli ' LINE 10, varlitikujte snimek I zatim otkucajte drugi deo programa i snimite ga sa SAVE C line 10

Sedam zlatnih pravila

Ovim smo se približili kreju ova škole akcionih igare ne Spectrumu Ipak pre nego što završimo poslednji čas r pozovemo vas da sami pitonete na posao nakoliko saveta koji će nademo se pomoći da vaše buduće vido-igre budu dinamičnije

Što više humora u komentarima koje računar ispisuje Neka korisnik programa vidi de igru ne treba shvetiti baš previše ozbiljno tpak treba paziti da sva šale izazivaju simpatiju prema kompjuteru

Ukoliko se pojavljuje situacija u kojoj je kompjuteiu potrebno duže vreme da se pripremi za sledeće akcije (npi da zamisli levinini na početku igie), pokušajte da stvorite utisak da kompjuter razmišlje Neka se na primer na ekranu s vremena na vieme smenjuju komentari u stilu „Samo što nisam spreman „ „Hmmm „ nije baš leko i slično Ovi komentari nipošto na smeju de budu isti pred svaku pertiju, jer će postati dosadni Pripremite više poiuka nego što je potiebno pa ih slučajno birajte pomoću RND

Uvek koristite karektere koje ste sami definisali umesto zvezdica ili nečeg sličnog Siguini smo da igreću neće ni malo smetati šio je bomba velike koliko i evion ili Mesec koliko i Sunce ako su istakunuti bitni detalji i ako se na prvi pogled vidi šta se dešave

Eksperimentišite što više sa bojema, koristite čak slučajna boje koje ipak morajo da budu kontiolisane u izvesnoj mari (eko PAPER i INK budu isto obojeni naće ee, videti šta se dešava) Promena boje pozadine neverovatno menja utisak o prizoru e košta samo jednu kreiku naredbu

Eksplozije su neobično efektne i na treba štedeti trud na njihovu realizaciju Treba samo paziti da eksplozije ne traju predugo jer one najčešće blokiraju sva druge događaje

Pošto ste zevišli igru utrošite još pola sata da smislite nešto što će biti originalno Predložili bismo de uvedete neki događaj koji je izuzetno malo vatovaten i dešave se ne primer, jednom u 200 partija Za korisnika će biti izuzetno prijetno iznenađenje da otkiije da igra koja mu je možde skoio dosadila ima još malih tajni

Iako ste tek završili prvi deo škole akcionih igaia na bajzaku razmislite o prelasku na mašinski jezik!

Dejan Ristanovic

Servis računara

„Galaktičari" u nevolji

Pre nego što vas obuzme panika, proverite još jednom štampane veze od video izlaza preko integralnih kola 74LS38, 74LS00, 74LS123, CD 4017, CD 4040 do 74LS93 i 74LS04. Uključite ponovo računar i pokušajte da na ekranu registrujete bar neki signal. Možda će to biti slika bez ikakvih znakova (slova ili brojeva), ali sa karakterističnim rasterom. Ako se i sada na ekranu ništa ne pojavljuje, ostaje vam mogućnost „hladnog lema" (postoji vizuelni ali ne i električni kontakt) na jednom od gore pomenutih integrisanih kola. Zagrejte lemilicu i osvežite lemove na ovim kolima. Ako i to ne pomogne, biće vam potrebna logička sonda da među ovim integralcima pronađete krivca.

Lociranje i otklanjanje greške je mnogo jednostavnije ako se na ekranu pojavi bar nekakva slika. Za brzo lociranje kvara pomoćr će vam tebele 1 sa pregledom integrisanih kola u čijoj okolini traba tražiti kvar.

Pročitajte još jednom

Ako još uvek niste dobili tasteturu ili kompletirali sve delove pročitejte sledeće savete pre početka rade

- dok na „oživite" računar, dovoljno je de ugradnte samo jednu memoriju 6116, kade računar proradi. leko ćete ugraditi i ostete e ako ete u nečemu drastično pogrešili, bar ćete sačuvati skupu memoriju.
- koristite RF moduletor firma ASTEC UM 1111 E 36, ovaj se moduletor napaje sa 12 volti, e raspored nožice je isti kao na štampanoj ploči. R16 zamanite kretkospojnikom a otpornik R15 izostevite za druge tipove modulatora potrebno je eksperimantalno odrediti vrednosti ovih otpornika.
- veš računar može u nevolji da redi r bez kondenzatore za blokadu Cn eko ih još niste nabavili probajte da oživite svoj računar i bez njih.
- posebnu pažnju obratite lemljenju najveći broj kvarove potiče uprevo odatle, prilikom lemljenja integrisanih kola napravite pauzu poste sveke naožrce dok se kolo na ohladi kade sa na njemu pojave nečistoće, vrh lemilica čistite vlažnom krpom.
- masku za tasteturu pre upotrabe treba malo obrediti ivice evih otvora za tastera malo sastružite turpijom i prvo postavite tastare na uglovima, e zatim po jedan u sredinr svakog rede.

Ako ste pažljivo sklepalr svoj računar i eko su vem sve komponente isprevne onda ćete prilikom prvog uključivanja imati sliku na ekrenu sa dugo željenim READY. Međutim, može se desiti da je slika iskrivljena r

SIMTOM	MOGUĆI UZROK
1. Ekran bez teksta i rastera	80A, 6116, 2732, 74LS04, 74LS83, 74LS74, 74LS123
2. Raster postoji ali nema teksta	2716, 74LS04, 74LS74, 74LS156
3. Slika jako iskošena	C3, C4, R12, R13
4. Ne dekodiju se određeni karakteri	74LS156, 74LS251
5. Karakteri se ispisuju ali nedostaju nek njihovi delovi ili su deformisani	Z80A, 2732, 6116, 74LS174
6. Paralelno se ispisuju prvi i peti red istih karaktarima	74LS174
7. Svi se karakteri ispisuju ali neki delovi nedostaju	74LS156
8. Slova presečena po vertikali a na BRK se javljaju crte na desnoj strani	74LS174
9. Nevezani karakteri na raznim delovima ekrana	Z80A, 74LS32, 74LS156
10. Pirlinčani raster po celom ekranu	Z80A, 4LS158
11. Promenljivi raster po celom ekranu	Z80A, 74LS156
12. Dijagonalni raster sa jednom vertikalnom strelom	Z80A, 74LS156
13. Nevezan tekst, ne prima komende	74LS38, 4LS74
14. Vertikalne šare bez teksta	74LS04, 74LS38
15. Crtice po celom ekranu	74LS38
16. Pored READY povremeno se javlja nevezi tekst	74LS38
17. READY na desnoj polovini ekrana zatim karakteri po celom ekranu	Z80A, 74LS74
18. Slika se kreće po vertikali	74LS04, 74LS123
19. Slika jako iskošena i kreće se po vertikali	C3, R12, 74LS123
20. Kosi tačkasti karakteri	74LS123
21. Par belih kosih crta	74LS123
22. Vertikalno postavljeni ili iskošeni karakteri	74LS93, CD4040
23. Vertikalno kretanje slike bez teksta	74LS93
24. Na PRINT MEM pokazuje manje memorije	6116,74LS156

da sa slova pojavljuju u više iskošenih redove. Pokušajte prvo da povećate kondenzetor C3 na oko 10 nF do 30 nF i otpornik R12 na 1 do 2 koma. Slike će početi da se ispravlja. Početak rede možete pomaretl smanjivanjem kondenzatora C4 do 50 nF i R1 do 15 koma.

Nepotuni karakteri

Umirili ste sirku i sade proveravate de ir se eveko elovo koje otkucete pojevljuje na ekrenu. Ako nekr taster ne reaguje odmah proverite njegovog susede sa vertikalna i horizontalne linrje na shemr. Ako ne reaguje čitev red proverite veze ka kolrna 74LS251 r 74LS156. Možda je te linrje u prekidu, ili nožrca kola nije dobro zalemljena. Posle ove mtervencrje tastetura će reagovetl rl je neisprevno kolo ka koma vodi linrje koje ne reaguje ne komande sa tasteture.

Na ekranu se mogu pojavrti r nepotpunr karakteri (recimo, slovo A bez horizontelna crte po sredinr). Grešku treba tražiti ne nožrcerna 4 i 5 kola 74LS174 rlr na nožrcerna 2, 3, 4, 5, 10, 11, 12 r 14 kola 74LS166, tj. na njihovom putu ka epromu 2716. U slučaju de su slove dvostruka, proverite nožice 6 i 7 kole 74LS174, kao i kole ka kojima ona vode. Loše zalemljene nožice 2 i 3 ovog kole daju deformisane i neprepoznatljiva karektere e prilskom na tester. BRK na desnoj strenr ekrena pojevljuju se vertikalne linije. Ako su karekteri presečenr po horizontali, proverite nožice 10 i 11 e eko sa karakteri ispisuju istovremeno u prvom r patom redu, odgovorne su nožice 14 r 15 kole 74LS174.

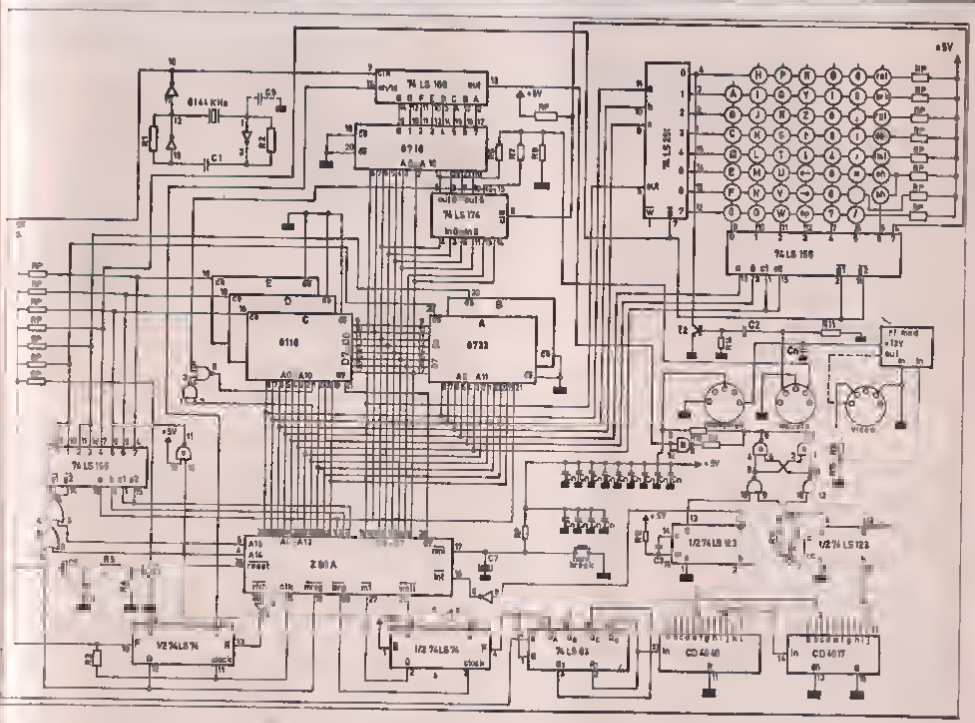
Neisprevno koto 74LS32 ili 74LS156 (u adresnom dekoderu) prouzrokoveće slike pirmčanog rastere ili će se na raznim delovima ekrena pojarti nevezeni kerekteri koji mogu brr i nestebilni. Ako se pojavi dijegonalni rester sa jednom vertikalnom štrelfom, nožica 15 kole 74LS156 ima kratak spoj sa masom. Isti se simptomr jevljaju i kad je neisprevan mikroprocesor Z80A.

Nevezani tekst

Nevezani karakteri eli sa ispisanim READY na početku prvog reda, jevijaju se ako nožrca 11 kole 74LS38 ima spoj sa masom. Ovej deo kole, kada je ono ispravno, prouzrokuje vertikalne šare na ekranu rlr nevezan tekst. Potpuno jed

Najzad je sve gotovo! Posle svih muka sa nabavkom delova, čekanjem tastature i čipova, vaš računar je, najzad, aktiviran. Uključili ste ispaljač, povezali kablove i počeli da pretražujete frekventna opsege na svom prijemniku, a na ekranu se ne pojavljuje ama baš ništa. Baš nemate sreće! Ne treba, međutim, gubiti glavu — to nije najgora stvar koja je mogla da vam se desi. Do sada je, osim toga, proradilo oko dva hiljada „galaksija", mnoge i u znatno naveštijim rukama nego što su vaše. Zašto biste baš vi bili izuzetak?

Računar „galaksija" odlikuje visoka pouzdanost u radu — do sada sa redakciji javilo samo nekoliko čitalaca, u svakom slučaju manje od deset koji nisu mogli sami da izađu na kraj sa svojim računarom, tako da praktično nemamo nikakvih serviserskih iskustava. Zato smo jedan od redakcijskih primeraka stavili „na muka", iznajmljujući na njemu, naravno po odredenom sistemu, sve kvarove koji su mogli da nam padnu na pamet. Računar je ovu torturu glatko preživeo, a rezultati su sumirani u tabeli i u ovom tekstu. Ako vam ni on ne pomogne da se izvučete iz nevolje, javite se redakciji. Vaš računar, budite sigurni u to, neće ostati samo gomila skupih čipova.



Električna shema računara galaksija — na shemi nema nikakvih izmena — jedino su obeležem izvodi na integrisanim kolima što znatno olakšava traganje za kvarovima

sa manifestuja prekid signala za osvežavanja memorija (koji inače ovde ima sasvim drugu ulogu) nožica 26 mikroprocesora, ako je neispravan Z80A nožica 3 i 4 kola 74LS04 ili nožica 13 kola 74LS74

Može se desiti da, neposredno posle uključivanja računara, na desnom delu ekrana dobijate READY, a da se odmah zatim pojave navezani karakteri po celom ekranu Grešku potražite na nožicama 2 3 ili 5 kola 74LS74 ili na nožicama 20, 24 ili 27 mikroprocesora

Vertikalno kretanje slika izaziva neka nepravilnost u lancu delitelja učestanosti kristala — kola 74LS93, CD4040 CD 4017 ili esm monostabilni multivibrator 74LS123. Tačne učestanosti signala na ovim kolima ne možete izmeriti bez specijalnih instrumenata (osciloskopa ili frekvencmetra) pa vam jedino preostaje da kola zamenite novim

Kad računar potpuno proradi ugradite preostale memorije 6116 i otkucajte PRINT MEM Ako računar pokaže da nemate onoliko memorija koliko ste ugradili integrisanih kola proverite nožice 4 i 5 kola 74LS 156 ili promenite mesta memorijama i opet istestirajte kapacitet memorija

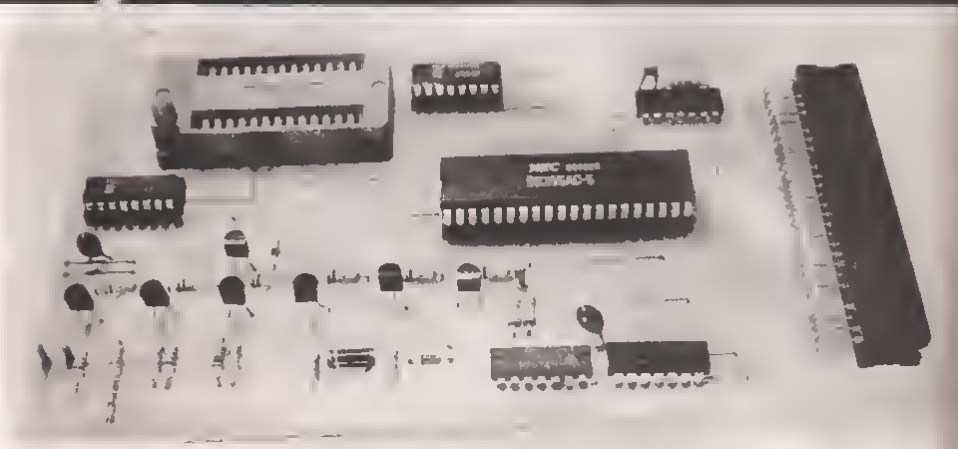
Sve navedene kvarove mnogo lakše ćete otkloniti ako imate mogućnosti da koristite direktan ideo ulaz u svoj televizor. Ili vam je na raspolaganju monitor U tom slučaju izbegavate greške koja unosi RF modulator

Tekst Miša Hedži Đorđević

Crteži Mira Tiodarević

programator eproma

računari u domaćoj radinosti



Nov način razmišljanja u konstruktorskom radu. Programator EPROM-a na kome su pored ROM-ova i generatora karaktera za „galeksiju" uspešno programirani i ROM-ovi za računare Spectrum i BBC može da uvede mnoga konstruktora i programera u krug potpuno novih ideja

Projekat programatora EPROM-a može sa nekoma učiniti kao evojevrsno preterivanja. Zar zaleta na postoji ništa zanimljivije i važnije? Na! Naravno, za ona koji na žele da sede skrštenih ruku i čekaju šta će im za samogradnju ponuditi najesen „Računari 3', „Galaksija" ili neki od stranih časopisa. Stvaralački rad u računarskoj tehnici teško je ostvariv bez jedne ovakve naprava. Mikroprocesor ja potpuno bespomoćan bez uputstava za rad koja dobija od ROM-a, a ROM možete imati jedino ako ga sami napravite. Uređaji za povezivanje računara sa periferijskom opremom, interfajsi za procesno upravljanje, višekanalni generatori tonova, uređaji za sintezu glasa i eva cetale računaraka eitnice koje život znače rade znatno jednostavnije ako je operativni program u nekom ROM-u nego ako sa učitava sa kaseta. I na samo to! Kakva je korist od neke nove rutina ili naredbe koja povećava snagu računara ako ona prilikom svake upotrebe mora iznova da sa učitava sa traka? Šta tek da sa radi kada se u biblioteci programa nakupi tridesetak novih naredbi i funkcija i etotinak korisnih mašinskih rutina?

Programator EPROM-a može očigledno, korisno da posluži i onim ljubitaljima računara koji nemaju nimalo konstruktorskih ambicija. EPROM zaleta, predstavlje najjednostavniji, najpouzdaniji, najtrajniji, pa, sasvim eigurno, i najelegantniji medijum za smeštanje sistemskog softvera, alternativnih programskih jezika i izuzetno važnih uslužnih programa (kao što ja, na primer, program za obradu teksta) koji se svakodnevno koriste. On se u računar uključuje III direktno, u predviđena podnožja, kao kod „galaksija" i BBC-ja III ja dovoljan sasvim jednostavan hardvar za povezivanje, tzv. EPROM loader, kao kod Spectruma. Za njegovo aktiviranje nije potreban nikakav dodatni softver (BBC) til je dovoljno nekoliko prostih maštnskih rutina (Spectrum). Prietup programu je, povrh svega, tranutan i, što je najvažnije, on sa nalazi u računaru čim ga uključite III čim, određenom naredbom, pozovete potraban program.

Kako radi

Rad sa programima iz EPROM-a je komforniji čak i onda kada imata disk-jedinicu! Na ovom svetu, medutim, ništa nija savršeno, pe nieu ni EPROM-I brisanja i upisivanje podataka u njih sa na vrši u računaru nego u epecijainim uređejima, kao što je ovaj čiju ehemu objavljujamo, a čitava procedura, naročito brisanje, može da potraja i nakoliko časova. Pošto sa programira jedanput a koristi bezbroj puta, ovaj nedostatak EPROM-a ima samo akademski značaj. Proizvođači memorija ga, ipak, ehvataju veoma ozbiljno i već najavljuju čipova koji sa pod naponom ponašaju kao RAM (može sa, dakle, po njima pisati i brisati do mile volje) a bez napona kao EPROM (čuvaju podatke beskonačno dugo i to ona koji se zateknu u njima u tranutku isključenja). Ipak, sasvim je eigurno da naredna deset godina na komarcijalnom tržištu pripada klasičnim EPROM-ima.

EPROM-I, eigurni amo, na predstavlje više ni iz daleka takvu zagonatku kao nekoliko meseci kada su u vreme pojave tekstova o računaru „galaksija" izazivali pravu uzbunu medu našim čitaocima. Međuvremenu je svaki gradite| „galaksije" imao prilika da ih upozna sasvim dobro, uveri se da sa na radi o nimalo nežnim čipovima. Malo teorije, ipak, neće biti na odmat.

EPROM predstavlja skraćenicu angleskog izraza „Electricelly Programmable Read Only Memory" i označava memorijsko koja sa programira električnim putem i koja sa podaci mogu samo čitati. Programira programiranja EPROM-a ja gotovo minalno jednostavna: svaki bajt sa upisuje pod visokim naponom TTL impulsom koji traja 50 milisekundi. Ovaj „visoki napon" kreće sa između 25 V (kod 2716) i 13 V (kod 27256). Programiranje na mora da teče uzastopno — može sa programirati bilo koja memorijska lokacija u bilo koja vreme — pojedinačno jedna za drugom ili sa preskok. Jedanput programiran, EPROM može sačuvati upisani sadržaj najmanja deset godina — eko sa na izloži aunčevoj energiji.

Računar „galaksija", bez sumnje, predstavlja veliki podsticaj i za programera i za samograditelje računara. On razobličuje kućni računar kao složen tehnički proizvod i ohrabruje ljubitelje računara da i sami ugrade ponaku ciglu u tu građevinu. Ozbiljniji programarski i konstruktorski rad na razvoju i, naročito, „doradama" računara (najomiljenija sportska disciplina domaćih hakerai) ne može se, međutim, ni zamisliti bez alatka koja sa naziva programator EPROM-a i koja omogućuje jednostavnu produkciju ROM-ova. Zato smo za treću samogradnju posle „galaksija" pripremili projekat univerzalnog programatora EPROM-a (adaptacija i dorada shema objavljene u namačkom časopisu MC) za računare „galaksija" i ZX Spectrum. Zašto baš za njih? Zato što su to dva najpopularnija računara na domaćem tržištu i zato što u njima ima najviše prostora za dorade. Uređaj programira najznačajnija savremena čipove EPROM-a — 2516, 2716, 2732, 2732A, 2764 i 27128 — i radi pod punom programskom kontrolom korisnika. Pomoću njega sa može iskopirati ROM bilo kog kućnog računara — od „galaksija" do BBC-ja — ili isprogramirati alternativni ROM sa nekim komercijalnim ili svojim sopstvenim programom.

Lična karta EPROM-a

tip	(kapacitet kilobajta)	potrošnja mirno/akt	brzina odziva
2716	2 Kb	25/100 mA	350 ns
2732A	4 Kb	35/125 mA	200 ns
2764	8 Kb	40/100 mA	200 ns
27128	16 Kb *	45/150 mA	200 ns

* Intel

sti ili nekom još intenzivnijem izvoru ultraljubičastih zraka. Broj ciklusa pisanje (impulsom) — brisanje (ultraljubičasti zraci) praktično je neograničen

I pored sva jednostavnosti procedura programiranja i brisanje EPROM-a može manje iskusnim konstruktorima izgledati krajnje opasno šta se dešava kada se kroz nežnu strukturu čipa propusti 25 V ili kada se ona izloži razornim UV zracima? Očigledno, upravo ono što je potrebno!

Memorijsku ćeliju u EPROM-u predstavlja MOS tranzistor sa dva gejta jeden je slobodan za prijem signala, a drugi obložan košuljicom silicijum-dioksida pošto na njega ništa nije vezano, najčešće se naziva plivajući gejt Silicijum-dioksid je izvanredan izolator — elektroni ne mogu ni da zadu napolje iz košuljice 'na plivajućem gejtu niti de udu unutra Kada se, međutim, na drejn dovede visok napon e na sloboden gejt nizak, tako da potencijalna razlika između njih iznosi 20 do 25 V, elektroni dobijaju dovoljno energije de probiju barijeru silicijum-dioksida i naelektrišu plivajući gejt U ovoj „košuljici ostaju zarobljani praktično zauvek — u svakom slučaju najmanje deset godina Ćelija se naelektrisanim plivajućim gejtom ponaša se kao *prazna* memorijska ćelija. Ultraljubičasto zračenje skebi izolaciona svojstva silicijum-dioksida Ako je dovoljno intenzivno ,košuljica će se toliko „razmekšati" da će elektroni „iscuriti kad da n nikada nije ni bilo — EPROM će biti obrisan i spreman za upis novog sadržaja Ćelija sa razelektrisanim plivajućim gejtom ponaša se kao *puna* memorijska ćelija Zato se u praznom EPROM-u u svakoj ćeliji nalazi logičko „i" , odnosno na svakoj adresnoj lokaciji broj 255 (heksadekadno FF) Prilikom programiranja u EPROM se, dakle, upisuju samo logičke nule, a prilikom brisanja te nule se vraćaju ne jedinice

Familije

Na tržištu se, trenutno, mogu naći dve nekompatibilne familije EPROM-a jedna nosi oznaku 27XXX i dolazi iz proizvodnih hala čuvenog intela, a druga 25XX i pripada Texas instrumentsu Kao i se mnogim drugim stvarima, Texas instrumenta ni ovde nije imao sreće intelova strandarde i Intelovu familiju, koje, povrh svega, ima i nešto više članova, prihvatilo je znatno više japanskih i evropskih proizvođača memorije I Ti je ostao gotovo usamljen. Jedan tip EPROM-a (2516) kompatibilan je, doduše, sa Intelovim 2716, ali su zato TMS 2532 i TMS 2564, sa drugačijim rasporedom pinova i nešto drugačijom procedurom programiranja, ostali sasvim izvan „serije" Njihova uloga sa, izgleda, svaie na to da piaše kupce visokom cenom i do besvesti kompliku ju univerzaine uredaje za programiranje EPROM-al

Naš programator EPROM-a uraden je preme intelovim standerdima On može da programira sve članove intelove tamilije — 2716 2732, 2732A 2764 i 27128 — celm najnovijeg 27256, jer se on pojavio kade je uredaj već bio završen iako, u principu, na bi bilo naročitih teškoće da se uredaj osposobi da programira i EPROM-a iz familije 25XX, ova pogodnost ipak nije ugrađena, jer bi programator previše iskomplikovaie, i softveraki i hardvarski Pošto su čipovi po inteiovom standerdu jeftiniji i uz to, pristupačniji od EPROM-a Ti, ovakva kompilkacija zaista nije bile potrebna Za one koji ne mogu bez familije 25XX, dajemo uporednu tabelu se rasporedom izvode (sl 1) i procedurom programiranje (sl 2) za EPROM-e iz obe familija Iz njih se može videti kako se na „geiskijinom programatoru EPROM-a mogu uz malo veštine i nekoliko hardvarskih rezova programirati i Ti čipovi

Svaki EPROM ima četiri kontroine funkcije — CS = izbor čipe, OE = dozvole izlaza, Vpp = programski napon i PGM = programski impuls — i, u zavisnosti od logičkih stanja na kontrolnim ulazima, pet rzziičitih modova rada mirovanje (stand by), čitanje, programiranje, provera i zadrške (inhibit) Procedurs za mirovenje čitanje i provaru je krajnje jednostavna I, rekio bi se, spontana, i otuda izbor koncepcije programatora treba da vodi računa prvenstveno o tome kako da se na najbolji način zadovolje kriterijumi za proceduru programiranja. (Zadrška se koristi samo kod istovremenog programiranja više EPROMA različitim sadržajem.)

Procedura programiranja ne bi, takode, bila vredna pažnje da upis jednog bajta, sa stanovišta mikroprocesora, ne traje čitavu večnost mikroprocesor drži adresu i podatak koji joj pripada stabilnim svega nekoliko stotina nanosekundi i odmah prelazi na sledeću, dok je EPROM-u za prihvatanje podatka potrebno čitavih 50 milisekundi. Zato okosnicu svakog programatora čini sklop koji „zaustavlja adresu i podatak dok se ovaj na upiše u EPROM i, razume se, obezbeduje programski impuls u trajanju od 50 ma Klasična koncepcije programatora EPROM-a zasniva se na tzv. osmobitnim „ječevima" (latch) — primitivnim memorijama koje su u stanju da „upamte" jedan jedini bajt. Njihov zadatak je da „zamrznu adresu i podatak u trajanju od 50 ma Uz izlazne batara, edrseni dekoder i još nešto maio hardvera, ovakva rešenja obavezno krasi i višepolni preklopnik za izbor tipa EPROM-a

PIO 8255

Sva svojstva koja se očekuju od ovakvih sklopova objedinjena su u jednoj posebnoj vrsti čipa poznatoj pod imenom PIO (Parallel input Output Port — grubo paralelna ulazno-izlazna kapija) Ova vrsta čipa je specijalno projektovana za dvosmernu paralelnu komunikaciju mikroprocesore sa periferijskim uređajima, pri čemu se njihova funkcija odreduja isključivo programskom kontrolom Programator EPROM-a, bez sumnje, predstavlja najjednostavniji uređaj koji pomoću PIO čipa može da se spregne sa mikroprocesorom Zašto se ta pogodnost ne bi iskoristila de se samo sa nekoliko čipova napravi uredaj profesionalnih mogućnosti i — karakteristika?

Svaka porodica mikroprocesora mora neizostavno, da ima i svoj PIO čip On se u Ziiogovoj familiji Z80 zove, jednostavno Z80 PIO Mada je namenjen za spregu se računarima izgrađenim oko procesora Z80 u ovom programatoru EPROM-a je ipak upotrebljen intelov čip iz familije 8085 se oznekom 8255AC-5 Zašto? Pored toga što je njegova i mašinska i programska sprega čak i sa ,tudim procesorima jednostavnija od sprega Z80 PIO sa svojim sopstvenim procesorom, Intelov PIO čip ima čitav jedan port više, što u ovoj primeni predstavlja presudnu prednost Da budemo do kraja jseni Z80 PIO se ne može koristiti u ovoj shemi bez ozbiljnijih hardvarskih izmena jer nema dovoljno izlaza de se ostvera sve potrebne funkcije!

Sprega PIO čipa 8255 sa mikroproceso-

rom (čitaj kućnim računarom) je veoma jednostavna. Pa ipak, ovom čipu je u Intelovom katalogu posvećeno tridesetak strana, što sasvim rečito govori o njegovoj fleksibilnosti i univerzalnosti primene. Iako bi detaljnija upoznavanja sa ovim čipom bilo, bez sumnje, od velike koristi, jer je nezamenljiv u interfejsima paralelnog tipa, ovoga puta se, eliom prilika, moramo ograničiti samo na najelementarnije stvari — dakle samo na ono što je od bitne važnosti za razumevanje rada programatora EPROM-a. Pažljiv konstruktor će, verujemo, iz takvog opisa izvući dovoljno praktičnih podataka i za mnoge druge primene.

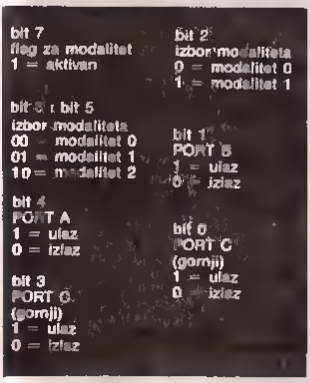
PIO čip 8255 sastoji se od tri porta — A, B i C — za dvosmerni prenos podataka i kontrolnog registra koji određuje način rada svakog od njih. Za upravljanje čitavim čipom dovoljno je svega nekoliko kontrolnih impulsa. CS prozivе PIO čip i omogućuje komunikaciju sa mikroprocesorom, RD naređuje PIO čipu da postavi podatak na DATA BUS mikroprocesora tako da ga ovaj prihvati i pročita s WR da upamti podatak koji preko DATA BUS-a upravo prima. Preko ulaza A0 i A1 prozivaju se pojedini portovi (vidi tabelu). Ako su, na primer, oba ulaza niska, biće aktivan PORT A, a eko su oba visoka kontrolni registar

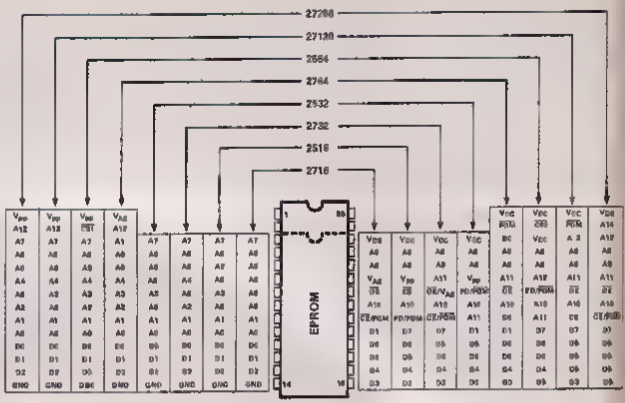
A1	A0	PORT
0	0	A
0	1	B
1	0	C
1	1	kontrolni registar

PIO čip 8255 može da se koristi u tri različite vrste rada

modalitet 0 svi portovi i ulaz i izlaz
modalitet 1 instraktivni ulaz i izlaz
modalitet 2 dvosmerni bus

Vrstom rada se upravlja potpuno programski. Za to su dovoljna samo dva instrukcije — podeci se upisuju pomoću mašinske III bajzik naredba OUT, e čitaju pomoću mašinske ili bajzik naredba IN. Da bi se odredila vrsta rada čitavog čipa i svakog porta u njemu, potrebno je, pra svega, prozvati kontrolni registar naredbom OUT i u njega upisati tzv. kontrolni bajt. Svaki bit u kontrolnom bajtu ima za PIO čip određeno značenja





Svi EPROM-i sveta. Uporedni raspored izvoda na svim tipovima EPROM-a koji se mogu naći na tržištu, naš programator razume se, nije dovoljno univerzalan da prihvati svaki od njih, ali to nimalo ne umanjuje njegovu vrednost

	2716		2732		2732A		2754		27128		2516		2532		2564	
	RO	PGM	RO	PGM	RO	PGM	RO	PGM	RO	PGM	RO	PGM	RO	PGM	RO	PGM
2	NC		NC		NC		A12		A12		NC		NC		$\overline{CS1}$	
	•	•	•	•	•	•	#	#	#	#	•	•	A	•	"0"	"0"
20	$\overline{CE}$/POM		$\overline{CE}$/$\overline{POM}$		$\overline{CE}$/$\overline{PGM}$		$\overline{CE}$		$\overline{CE}$		PO/POM		A11		A11	
	"0"	⎍	"0"	⎍	"0"	⎍	"0"	"0"	"0"	"0"	"0"	⎍	#	#	#	#
22	$\overline{OE}$		$\overline{OE}$/V_{pp}		$\overline{OE}$/V_{pp}		$\overline{OE}$		$\overline{OE}$		$\overline{CS}$		PO/$\overline{PGM}$		PO/$\overline{PGM}$	
	"0"	"1"	"0"	+25	"0"	+21	"0"	"1"	"0"	"1"	"0"	"1"	"0"	⎍	"0"	⎍
23	V_{pp}		A11		A11		A11		A11		V_{pp}		V_{pp}		A12	
	"1"	+25	#	#	#	#	#	#	#	#	1	+25	"1"	+25	#	#
26	V_{CC}		V_{CC}		V_{CC}		N.C.		A13		V_{CC}		V_{CC}		V_{CC}	
	+5	+5	+5	+5	+5	+5	•	A	#	#	+5	+5	+5	+5	+5	+5
27	NC		NC		NC		$\overline{PGM}$		$\overline{PGM}$		NC		NC		$\overline{CS2}$	
	•	•	•	•	•	•	"1"	⎍	"1"	⎍	•	•	A	•	"0"	"0"
V_{pp}	25 V		25 V		21 V		21 V		21 V		25 V		25 V		25 V	

. . nije važno
= dinamički ulaz

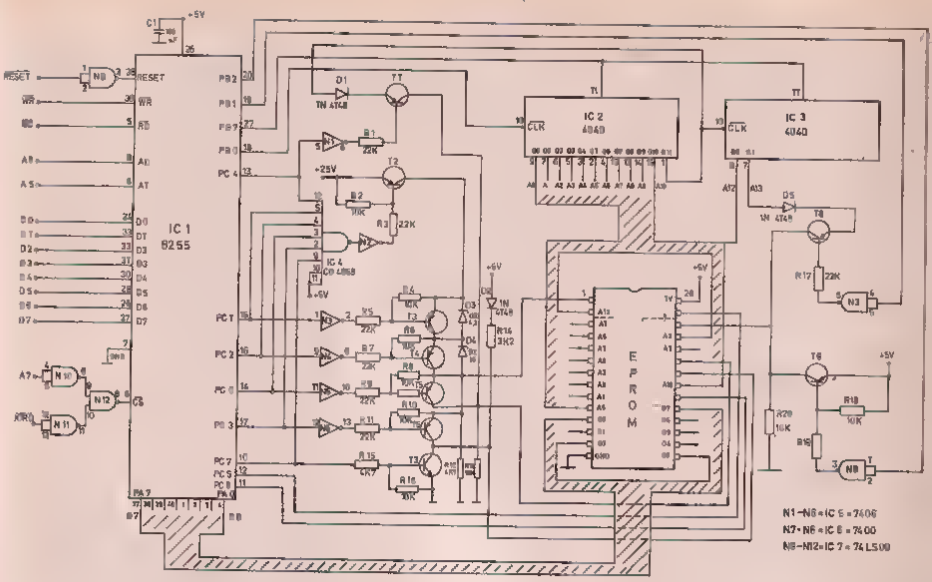
Procedura programiranja i čitanja EPROM-a. Uporedni raspored izvoda sa kontrolnim impulsima i logičkim nivoima za različite tipove EPROM-a, kod EPROM-a sa 24 izvoda broj nožice je za dva manji od onoga koji je naveden (nožica 1 dolazi na nožicu 3 s nožica 24 na nožicu 26 u podnožju)

Upisivanje kontrolnog bajta u kontrolni registar naziva se inicijalizacija PIO čipa i ona se na može izbeći — bez nje PIO čip naprosto na zna šta treba da radi! Nakon ovoga PIO čip ostaje u određenom modalitetu, e portovi u određenoj vrati rada, sve dok se ponovo na pozove kontrolni registar i u njega na upiše novi kontrolni bajt.

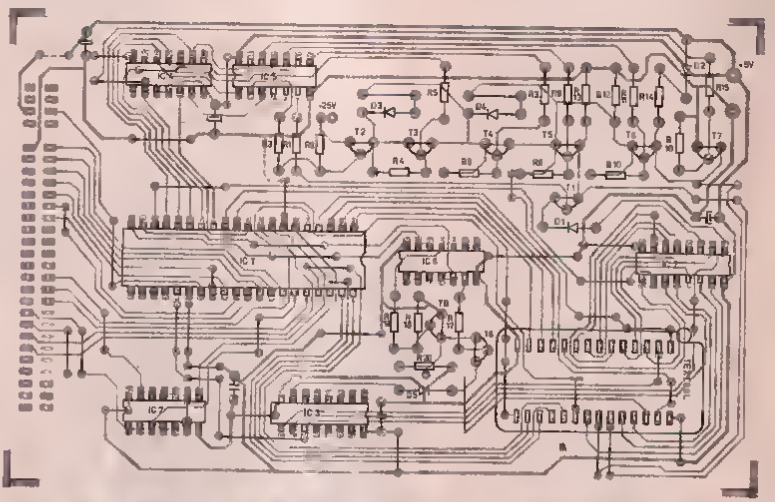
Pretpostavimo da želimo da PIO čip radi u modalitetu 0 i da sva tri porta funkcionišu kao izlazna. Ako su kontrolni ulazi A0 i A1 vezani na najniže adresne linije s CS na linij a2, port A će imati adresu 0 (binarno 000 — A2=0, A1=0 i A0=0), port B — 1 (bin 001), port C — 2 (bin 010), a kontrolni registar 3 (bin 011). Kontrolni bajt treba da ima vrednost 128 (bin 10000000).

Bajzik naredba OUT 3, 128 (= prozovi kontrolni registar i upiši u njega broj 128) izvršiće inicijalizaciju čipa i odrediti vrstu rada — setovati fleg za modalitet (bit 7 = 1), „uključiti" modalitet 0 (bit 6 = 0, bit 5 = 0 i bit 3 = 0) i postaviti PORT A (bit 4 = 0), PORT B (bit 1 = 0) i PORT C (bit 3 = 0 i bit 0 = 0) u izlazni modalitet.

Mini-program koji bi nečim uposilo sve ki od portova mogao, na primer, da izgleda ovako
OUT 3, 128 — Izvrši inicijalizaciju



Savršeno jedinstvo mašinskog i programskog dela (hardvera i softvera Električna shema programatora EPROM-a uređaja



Montažna shema Kod složenijih uređaja na jednostrukoj štampi kratkospojnici predstavljaju uobičajenu pojavu ima ih i ovde podosta ali ne toliko da ispora samogradnju ili pokvara estetski izgled programatora. prilikom programiranja čipovi sa ne nožice postavljaju se u podnožja kao što je prikazano isprekidanom linijom na električnoj shemi

KOMPONENTE

Integrisana kola

IC 1	8255AC-5	
IC 2, IC3	CD 4040	
IC 4	CD 4050	
IC 5	SN 7405	
IC 6	SN 7400	
IC 7	SN 74LS00	

Tranzistori

T1-T6 Te
T7
T8

Diode

D1 D2 D5 1N 4148
D3 BZ 1, 3
D4 BZ 15

Otpornici

R1, R5-R7 2
R2 1
R3, R4, R8, R9, R11, R12, R13, R14, R15, R16, R17, R18, R19 22 kома
R10 4k7
R20 10K
R21 2k2
R22 1K
R23 680 koma

Kondenzatori

C1-C4 100 nF (blok)
C5 1 uF (tantal)

Razno

ZIF 28 nožica (Textool) 1
40 nožica 1
16 nožica 2
14 nožica 1

Konektor

2 × 22 razmak 3.95 mm (galaksija)
2 × razmak 2.5 mm (Spectrum)

```
 10 REM *****HEXLOADER******
 20 DEF FN h$(a)=CHR$ (INT (a/1
6)+48+7*(a>159))+CHR$ (a-16*INT
(a/16)+48+7*((a-16*INT (a/16))>9
))
 30 DEF FN h(h$)=CODE h$-48-7*(
h$>"9")
 40 INPUT "STARTNA ADRESA ";s
 45 INPUT "ZAVRSNA ADRESA ",f
 50 FOR n=s TO f STEP 8
 60 PRINT n,"  ";
 65 INPUT a$
 70 IF a$="S" THEN STOP
 75 PRINT a$
 80 FOR b=0 TO 7
 90 LET z=FN h(a$)*16+FN h(a$(2
))
 100 POKE (n+b) z
 110 LET a$=a$(4 TO )
 120 NEXT b

 130 NEXT n
 140 STOP
 499 REM *****HEXLIST*****
 500 INPUT "STARTNA ADRESA ",X
 510 INPUT "ZAVRSNA ADRESA ",Y
 520 CLS
 530 PRINT X,"  ";
 540 FOR A=1 TO 8
 550 LET A$="  "
 560 LET B=PEEK X
 570 LET A$=CHR$ (INT (B/16)+48+
7*(B>159))+CHR$ (B-16*INT (B/16)
+48+7*((B-16*INT (B/16))>9))
 580 PRINT A$," ";
 590 LET X=X+1
 600 IF Y=X THEN STOP
 610 NEXT A
 620 PRINT
 630 GO TO 530
```

Program 1
Program za unošenje mašinskog kode (tzv hex-loader). Bajtovi se ne unose jedan po jeden kako je to uobičajeno nego red po red — ENTER dakle treba pritisnuti tek na kraju reda

```
  10 LLEAR 24999  LOAD "EPROMLIF"
CODE 25000 724
  20 PRINT AT 5,6 "PROGRAMATOR E
PROMA"
   30 PRINT AT 11 10  1 = /71n"
   40 PRINT AT 12 10  2 = 2732
   50 PRINT AT 13 10 "3 = 27326"
   60 PRINT AT 14 10 "4 = 2764"
   70 PRINT AT 15 10 "5 = 27128
   80 INPUT "TIP EPROMA" A
   90 IF A=0 OR A>5 THEN PRINT AT
 21 0 "NETACAN PODATAK    GO TO
20
  100 POKE 23300 A
  110 CLS   PRINT AT 11   "1 = PR
OGRAMIRANJE"
  130 PRINT AT 12 7 "2 = KOPIRANJ
E"
  140 PRINT AT 13 7 "3 = LISTHITE
"
  145 PRINT AT 15 7 "4 = KRAJ
  150 INPUT C

   160 IF C=4 THEN PRINT AT 21 0
KRAJ"
  170 CLS   INPUT "POCETNA ADRESA
 U RRM-u" U
  180 POKE 23296 U-256*INT (U/256
)   POKE 23297 INT (U/256)
  190 IF C=1 THEN GO TO USR 25000
  200 IF C=2 THEN GO TO USR 25155
  210 IF C=3 THEN GO TO 230
  220 LET A$="0123456789ABCDEF"
  230 LET X=INT (PEEK U/16)  LET
Y=PEEK U-16*INT (PEEK U/16)
  240 PRINT U TAB 7 A$(X+1) A$(Y+
1)
   250 LET U=U+1
   260 GO TO 230
```

Program 2
Program u bejziku ispisuje meni i prosleđuje parametra mašinskom programu — ovo rešenje je prihvaćeno da bi se izbeglo pisanje novog aditora

Program 3
Mašinski kod

```
HEXLIST

25000 CD EC 62 01 C2 63 CD 34
25008 63 CD 58 63 DD 7E 01 D3
25016 3F DB 1F FE FF 28 F4 CD
25024 9D 62 20 F5 DD 7E A0 D3
25032 3F CD 10 63 3E 80 D3 7F
25040 CD 29 63 DD 7E 02 D3 3F
25048 7E FE FF 28 18 D3 1F DD
25056 7E 03 D3 3F F3 3E 92 0E
25064 DB 00 20 FD 3D 20 F8 F8
25072 DD 7E 02 D3 3F CD 0D 62
25080 20 DE CD 10 63 CD 24 63
25088 CD 23 63 DD 7E 01 D3 3F
25096 DB 1F 46 80 20 25 CD 0D
25104 62 20 F5 01 05 64 CD 34
25112 63 CD 56 63 CD 5F 63 01
25120 14 00 C9 01 D8 63 CD 34
25128 63 CD 58 63 CD 5F 63 01
25136 14 00 C9 01 10 64 CD 34
25144 63 CD 58 63 CD 5F 63 01
25152 14 00 C9 CD EC 62 01 95
25160 63 CD 34 63 CD 58 63 DD
25168 7E 01 D3 3F DB 1F 77 CD

25176 20 35 3A 20 36 33 20 43
25184 44 20 35 46 20 36 33 20
25192 30 31 20 20 20 32 35 31
25200 32 30 3A 31 34 20 30 30
25208 20 43 39 20 30 31 20 44
25216 33 20 36 33 20 43 44 20
25224 33 34 20 20 20 32 35 31
25232 32 36 30 36 39 20 43 44
25240 20 35 30 20 35 33 20 43
25248 44 20 35 46 20 36 33 20
25256 30 31 20 20 20 32 35 31
25264 33 30 3A 31 34 20 30 30
25272 20 43 39 20 30 31 20 31
25280 44 20 36 34 20 20 20 31
25288 33 34 20 20 20 32 35 31
25296 34 34 36 35 33 20 43 44
25304 20 35 30 20 36 33 20 43
25312 44 20 35 46 20 36 33 20
25320 30 31 20 20 20 32 35 31
25328 35 32 3A 31 34 20 30 30
25336 20 43 39 20 43 44 20 45
25344 43 20 30 32 20 30 31 20

25352 30 30 20 20 20 32 35 33
25360 33 35 3A 32 30 20 34 33
25368 20 33 39 20 32 30 20 34
25376 33 20 34 34 20 32 30 20
25384 34 35 20 20 20 32 35 33
25392 34 34 3A 34 33 20 32 30
25400 20 33 36 20 33 32 20 32
25408 30 20 33 30 20 33 31 20
25416 32 30 20 20 20 00 50 16
25424 80 50 CD 6E 0D D1 E1 C9
25432 01 96 00 CD 3D 1F C9 01
25440 9D 64 CD 34 63 01 53 00
25448 CD 34 63 CD 40 63 CD 58
25456 63 C9 41 01 97 47 64 04
25464 05 58 90 5E 1E 64 04 05
25472 58 90 5C 1C 64 04 05 70
25480 80 34 14 64 04 05 70 80
25488 34 14 02 02 03 16 09 00
25496 4B 4F 50 49 52 41 4E 4A
25504 45 09 16 09 06 4B 4F 50
25512 49 52 41 4E 4A 45 20 47
25520 4F 54 4F 56 4F 16 00 0A

25528 43 45 4B 53 55 4D 2A 3D
25536 20 0B 16 09 09 50 52 4F
25544 47 52 41 4D 49 52 41 4E
25552 4A 45 0B 16 07 06 12 01
25560 45 50 52 4F 47 4D 20 4E 49
25568 4A 45 20 50 52 41 5A 41
25576 4E 21 0B 16 09 04 53 54
25584 41 56 49 20 45 50 52 4F
25592 4D 20 55 20 50 4F 44 4E
25600 4F 5A 4A 45 0B 16 09 00
25608 50 52 4F 47 52 41 4D 49
25616 52 41 4E 4A 45 20 55 53
25624 50 45 4C 4F 07 16 07 03
25632 12 01 50 52 4F 47 52 41
25640 4D 49 52 41 4E 4A 45 20
25648 4E 49 4A 45 20 55 53 50
25656 45 4C 4F 21 0B 16 A9 04
25664 49 5A 56 41 44 49 20 45
25672 50 52 4F 4D 20 49 5A 20
25680 50 4F 44 4E 4F 5A 4A 41
25688 0B 16 0A 02 49 20 50 52
25696 49 54 49 53 4E 49 20 42

25704 49 4C 4F 20 4B 4F 4A 49
25712 20 54 41 53 54 45 52 12
25720 01 20 0B 00 00 00 00 00
```

Komponente

PIO čip 8255 i podnožje za EPROM predstavljaju dve najozbiljnije stavke u troškovima za gradnju programatora. Čip će do kraja epilia koštao u Engleskoj svega 3—4 funte a u Nemačkoj petnaestak maraka. Ne bi preko noći skočio na 19 funti (!) i prako 30 maraka. Podnožje je, takođe, veoma „paprano" oko 8 funti u Engleskoj, odnosno 35 maraka u Nemačkoj. Unapred se odricite ideje da ovo podnožje zamenite klasičnim, jako će stotinak puta jeftinije. Radi se o specijalnom podnožju koje se proizvodi upravo za ovakvu vrstu uređaja. Postavljanje čipa u njega je veoma jednostavno jer su otvori dovoljno veliki a kontakt se ostvaruje mehaničkom kočnicom. Na tržištu se može neći nekoliko tipove sa približno istom cenom, ali je ubedljivo najbolji onaj koji na sebi nosi oznaku „Textool".

Cene ostalih čipova nisu ni iz daleka tako šokantne — ukupan iznos ne bi trebalo da prade 35 funti sa poštanskim troškovima. Štampano kolo ne koma takođe ne treba štedeti: može se nabaviti preko redakcije „Galaksije" po ceni od *verovatno oko* 1000 dinara. Na ispravljeć će otići otprilike ista suma tako de kompletan uređaj ne bi smeo da košta više od 9000 dinara. Cene ovakvih uređaja kreće se u Engleskoj između 60 i 70 funti — uštede je dakle, više nego očigledna. Pošto se očekuje da će se tržište smiriti u drugoj polovini godine i cene krenuti blago nizbrdo — veruje se da se one više nikada neće vratiti na staru meru — čitava kalkulacija može da se pokaže i znatno povoljnijom.

OUT 0, 255 — uključi sve izlaze na portu A (255 = 11111111)

OUT 1, 0 — isključi sve izlaze ne portu B (0 = 00000000)

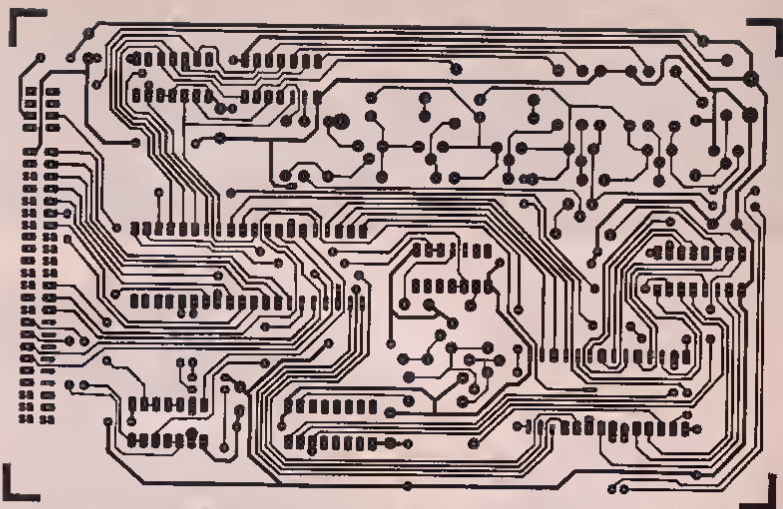
OUT 2, 129 — uključi prvi i poslednji izlaz na portu C (129 = 10000001)

Portovi, e suštine i jeste u tome, neće promeniti stanje sve dok mikroprocesor novom OUT instrukcijom ne uputi novi zahtev i novi — podatak. Port, drugim rečima, može da ostane „zamrznut" beskonačno dugo, e da to nimalo ne omata mikroprocesor — umesto da čeka da periferijski uređaj završi posao i vrti se u mrtvoj petlji, on može de radi ne drugim zadecima.

Adrese

U našem projektu za povezivanje PIO čipa i aktiviranje portova primenjene se nešto drugačije adrese. PIO čipu je, razume se, svejedno koja ge adresa poziva (koje će adresne linije biti vezane na CS, A0 i A1) ali ne i mikroprocesoru, pe ni računerima „galaksija" i Spectrum, iz jednostavnog razloga što ta adresa ne sme biti u memorijskoj, već se mora nalaziti u tzv ulazno-izlaznoj (I/O) adresnoj mapi iz te mape se uzimaju adresa za rad se svim periferijskim uređajima, pe se mora uzeti i adresa programator EPROM-a. Procesor Z80 u ovoj mapi ima semo 256 adresa (A0—A7), a toliko ih ima i računer „galaksija". Spectrum, doduše, i ovde ima 65536 adresa, ali je najbolje koristiti adresne linije A5, A6 i A7, jer ih sem računer ni za šte ne koristi. Ove adresne linije, dakle odgovaraju u svakom slučaju. Kod Spectruma velja zadovoljiti još jedan usiov: adresne linije A0—A4 moraju biti ne visokom nivou da bi

Štampano kolo u razmeri 1 1



zaštitila neke već postojeće adrese u I/O mapi.

Za prozivanje PIO čipa odabrana je adresna linija A7 i to za uslov kada se nalazi na niskom nivou. Ona, sama za sebe, međutim nije dovoljna PIO čip sme da se aktivira samo kad se mikroprocesor obraća svojoj I/O mapi. On to čini jedino kada izvršava IN ili OUT instrukciju. Da bi periferijski uređaj uopšte znao da mu se obraća, mikroprocesor mora u tom času da spusti nisko na svom izlazu IORQ (= tabi kažem) Logičnost tavan aklop sa NI kapijama omogućuje mikroprocesoru da orozove PIO čip samo ako su A7 i IORQ u isto vrame na niskom logičkom nivou. Ova situacija je ispunjena u 128 slučajeva (sve adrese od 0 do 127), ali to nija ni od kakvog značaja jer u tom času ne radi nijedan drugi perifarijski uređaj i ne može da doda ni do kakva konflikta. Apsolutno prozivanja PIO čipa jedinstvenom adresom, samo bi bilo trebno iskomplikovaio uredaj. Na konkretne ulaze A0 i A1 vezane su adrese A6 i A7 (A0 = A6, A1 = A5 — logičnija bi bila A0 = A0, A1 = A1, ali ovo malo pojednostavljuje shemu.) Pošto sa prozivanjem bilo kog čipa mora istovremeno da buda prozvan i taj čip, u adresu mora da ude i taj uslov

A7	A5	A4	A3	A2	A1	A0	dakadno adresa	
0	0	1	1	1	1	1	31	PORTA
0	1	1	1	1	1	1	63	PORT C
0	1	1	1	1	1	1	93	PORT B
1	1	1	1	1	1	1	127	kontrolni registar

Pošto između programatora EPROM-a i mikroprocesora u računaru na postoji nikakva interakcija (, handshaking — signali između mikroprocesor i PIO čip razmenjuju samo to da li su završili određenu operaciju ili ne) PIO čip radi u modalitetu 0 oba porta A pranose se podaci koji se upisuju u EPROM ili čitaju iz njega prako port C impulsi i naponski navoi koji brinu o procesuri programiranja, verifikacija i čitanja za različite tipove EPROM-a, a preko porta B impulsi za resetovanje i okidanje programskih brojače, kao i za elaktronski prekidač koji ne nožicu 24 dovodi ili napajanje (do 2764) ili adresnu liniju A13 (27128). Očigledno je da portovi B i C uvek rade u izlaznom a port A prilikom programiranja u izlaznom, a prilikom čitanja u ulaznom modalitetu. Potrebna su dakle samo dva kontrolna bajta

128 — modalitet 0, port A izlaz, port B izlaz port C izlaz

144 — modalitet 0, port A ulaz, port B izlaz, port C izlaz

Adrsairanja memorijskih iokacija u EPROM-u prilikom čitanja i programiranja izvedeno je, takode na krajnje neobičan način. Bilo bi logično da se taj posao poveri mikroprocesoru — edresiranje bi se tada odvijalo automatski. Tada bi, međutim, bila potrebna još dve „leća" za privremeno „zamrzavanja adresa. Štampano kolo bi se toliko iskomplikovaio da bi najverovatnija, bila potrebna dvostruka štampa, a programiranja postalo znetno složanija. Zato je u programator ugraden poseban programski brojač sa dve čipa 4040 (brojač sa modulom 16 — dali sa 4096). Ovakva koncepcija je alegantna i u mašinskom i u programskom smislu, ali ima i jednu alabu tačku: ako se EPROM ne programira aukoselvno, od početka do kraja, nego na preskok, traba dopisati petlja za okidanja brojača u prazno uvek kada se prsskače neka adrese.

Hardveraka koncepcije programatora EPROM-a omogučava potpunu programsku kontroiu avaka funkcija prilikom čitanja, tsetiranja programiranja i verifikacija EPROM-a i, samim tim, ostavlja korisniku potpuno odrešena ruke da odredi funkcionalnu koncepciju i prilagodi uredaj svojim videnjima i svojim potrebama. Opisanom hardverakom rešenju nedostaja, možda, samo jedna atvar: sopstveni ROM sa programom za rad uredaja koji bi se „aktivirao" čim se programator priključi na računar. Tima bi sa izbegio dosadno učitavanja programa sa traka avaki put kada traba isprogramirati neki EPROM, ali bi se programator ipak nepotrebno iskomplikovao. Rad sa programatorom ja, inače, u svemu ostalom veoma komforan: nakon startovanja programa dovoljno ja uneti tip EPROM-a i početnu adresu u RAM-u. Sve ostalo se dešava gotovo samo od sebe.

Programator EPROM-a, mada mu cena nija mala, spada u onu vratu uredaja koja sa ispiačuje praktično čim proradi i po tome ja u kompjuterakom avetu potpuno jedinatven. Razlika u ceni programe na kaseti i u EPROM-u su toliko draztične (prazan EPROM 27128 košta oko 25 a napunjan i do 60 funtil) da se programator ponakad isplaćuje samo jednim programiranjem! Ovakvi uredaji, međutim imaju i jednu mnogo dragoceniju vrednost: oni u programarakom i konstruktorskom radu otvaraju vrata za sasvim nove idaje.

Jova Regasek

Ovo nije sve!

U „RAČUNARIMA 3"

SOFTVERSKA KONCEPCIJA I PRAKTIČNA REALIZACIJA PLUS

- **Kompletan program za računar „galaksija" sa sledećim opcijama**
  - **— testiranje epproma i . . .**
  - **— parcijalno testiranje između dve adrese**
  - **— kopiranje EPROM-a u RAM i . . .**
  - **— . . . parcijalno kopiranje između dve adrese**
  - **— programiranje EPROM-a iz RAM-a i . . .**
  - **— . . . parcijalno programiranje između dve adrese**
  - **— verifikacija i . . .**
  - **— . . . verifikacija između dve adrese**
- **usavršeni program za računar Spectrum sa opcijama za testiranje, kopiranje, programiranje i verifikaciju bit po bit**

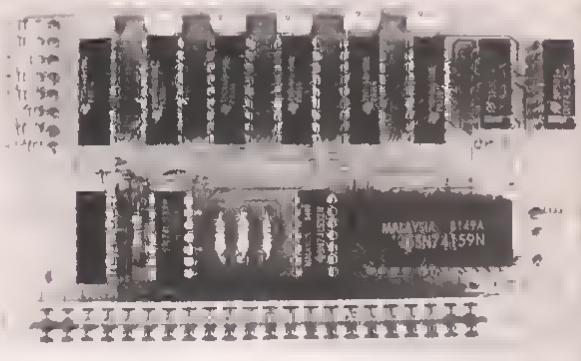
memorija za „galaksiju“

„Galaksija 48 K“

Da bi jeden računar funkcionisao kao skladna celina, svi njegovi sklopovi moraju da budu primereni jedni drugima. Moćan operativni sistem ne može da dođe do izražaja sa metom rednom memorijom (računaru QL nije izgleda dovoljno ni 128 K!) niti će velike memorije biti dobro upošljene sa skromnim opsrativnim sistemom. Kolike je memorije onda, primerena računere „galaksije ? Skeptici bi rekli da je on već sada ima dovoljno, eli u to ih može razuveriti svaki vlesnik „galaksije koji je pokušao da na svom računeru napiša barem jeden ozbiljniji program u bajziku. Oni koji tvrde da će se „gataksije sa 54 K redne memorije ,osećati kao u tudem odetu koja joj je uz to za In broje veće verovatno su nešto više u pravu. Učitavanja programa u toliku memoriju stažemo se može da potreja čitavu večnost. Nije međutim sve ni u brzini učitevanje čak i eko ona na bi mogle da se poboljša

Obilne memorije možde izgteda nezgrapno uz računer ‚galeksije – eli ona u mnogome može da nadoknedi skromnije osobine operativnog sisteme. Domeći ljubiteji računara, koji silom prilike nabavljaju mešine koje su znatno ispod njihovih potrabe, u doredeme operetivnog sistema često vide jedini izlez. Operetivni sistem „galeksije je savršeno otvoren ze ovakve dorede — računaru nedostaje samo malo više memorije. Veća memorije pored toge predstavlja osnovni uslov za hardverska poboljšanja računara kao što je na primer finija grefike i naročito za programski zamah. Ko će se prihvatiti da napiše „ozbiljan“ program za „galaksiju“ kad se sa 6 K radne memorije ne mogu pisati takvi programi?

Iako tranutne potraba, po pravilu, ne mogu de budu pouzdano merilo ze veličinu memorije jednog računere — nju će diktirati razvoj i hardvare i softvera koji za sade niko ne može de predvidi — svaki vlesnik „galeksije će očigledno moratl sem de odredi svoju meru. Uz one kilobajte koji već postoje u računeru 16 K. sasvim sigurno, predstevljeju najrazumniju količinu. Ako vreme pokaže (verovatno i hoće eli ne teko brzo) de je i to nedovoljno memorije se kasnije može jednostevno, uz dodatek samo dva ili četiri čipe proširiti za još 16 ili 32 K. Od potrošačke groznice niko nikade nije imao koristi: pe nečete ni vi i nemojte se plašiti de kasnije nečete neći potrebne čipove — doći ćete do njih sigurno i to verovatno znetno jeftinije. Realno 32 K predstavlja gornju razumnu granicu



„Galeksije 48 Memorijsko proširanje povećava radnu memoriju računara „galeksija u zavisnosti od broja ugrađenih čipova: ze 16 32 ili 48 K zehvaljujući fleksibilnom adresnom dekoderu proširanje se može pomerati po memorijskoj mapi u skokovima od po 4 K

Statičke ili dinamičke?

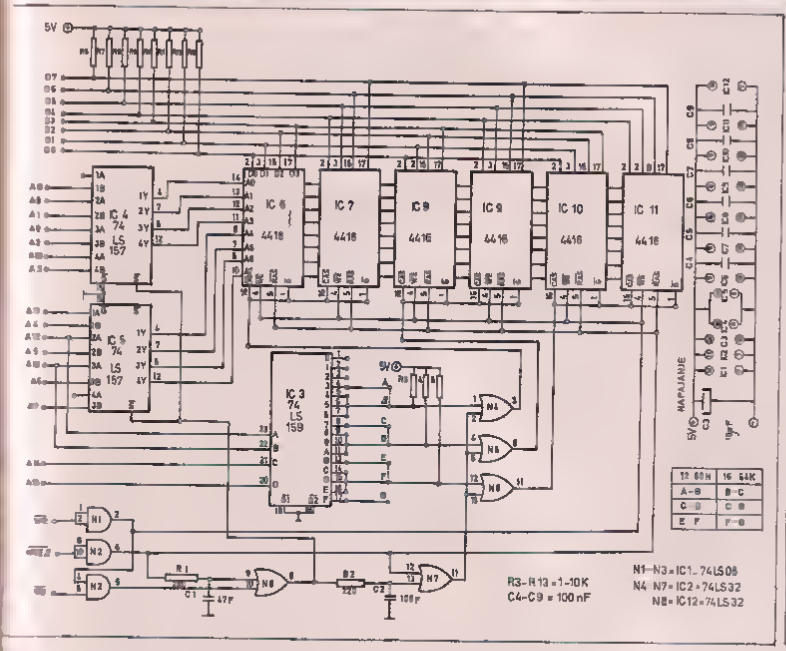
Ako je u tranutku kede je koncipiran računer geleksija i moglo de bude dileme oko tipe čipa koji će se koristiti u radnoj memoriji, sada nedoumici nema ni najmanje mesta. Statičke memorije koja se ugrađuju u „galaksiju“ toliko su poskupele na svatskom tržištu de ih preporučujemo samo u izuzetnim prilikame i — mikronskim količinama. Preme trenutnom peritetu dinamički kilobajt je gotovo osam (!) puta jeftiniji od statičkog što praktično znači de se za „galaksijinth“ 6K u osnovnoj verziji može dobiti gotovo 48 K u „dinemičkim“ bajtovima! Na izbor statičkih mamorije za osnovnu verziju „galaksije najviše je utica-la bojazan de dinamička memorije mogu konstruktorima de zegorčaju život više nego što je to nužno. Kakva je razlike izmedu statičkih i dinamičkih memorija i de ti je zebnja konstruktora „galaksije bile opravdena?

Memorijska ćelija u statičkoj memoriji izgradena je ne bazi tzv. flip-flope — dosta složenog elektronskog sklope koji se ostvaruje sa pet-šest tranzistora. Jednom upisan sadržaj ostaje u ovakvoj ćeliji sve dok se preko njega ne upiše novi ili dok se memorija ne isključi iz struje. Svi sklopovi potrebni za rad sa ovom vrstom mamorije ugrađeni su u sam čip. pe je procedura upisivanje i čitanja podeteka potpuno spontena. Statičke memorije su međutim veoma glomazne — ćole značajnije memorijske konfiguracija zahtevele bi mnoštvo čipova i ogromno štempeno kolo. Proizvodačl me je do sada pošlo ze rukom de u jeden čip smeste najviše 8192×8 ćetije (osam ki bajta) eli tekav čip košta 50 dolera!

Kod dinamičkih memorije ćelije je izgdena na bazi kondenzetora. Pošto zauzi ju manji prostor od ćelije se tranzistori kondenzatori obezbeđuju znatno veću stinu pakovenje i — menju cenu po ba Mikrometarske dimenzije međutim po ča ze sobom i mikrometerski kapacitet pat do deset stotih delova pikotaredal — se ćelija brzo prazni i njen sadržaj nepovratno gubi. Najbolji tipovi dinami memorija mogu da pamte podetak na četiri milisekunde. Zato se njihov sad more stalno obnevljati — to se post tekozvenim osvežavenjem. Osveža dinemičkih memorije koja uteruje streh kosti neobaveštenim tjubitelijima računa predstevlje trivijelan proces koji pro Z80 obavlje automatski ne kreju sva mešinskog ciklusa. Osvežavanje se post prozivenjem memorijskih ćelije prazno

Druga specifičnost dinamičkih me ja međutim nije toliko bezazlene i zaht i nešto malo dodatnog hardvera i odr proceduru ze upisivenje i ščitevanja p teka. Memorijske ćelije u dinemičkoj moriji su organizovane u redove i kolo eli je direktan pristup omogućen sa redovime. Zeto se adresa ne saopšt memoriji odjednom nego — kao i kad mobitih procesore — u dvaput. U m riju se najpre upisuje adresa za redove potom — u istom mešinskom ciklusu! — za kolone. Ćelije tek tada postaje spre de primi ili oda svoj sadržaj. Nešto kompilkovanije procedure rede i zavis

Radna memorija, bez sumnje, predstavlja jedno od osnovnih merila snage jednog računara. Bez obzira na snagu i kapacitet njihovog operativnog sistema, u savremenim kućnim računarima se za memorisanje, po pravilu, koristi sav prostor koji dopušta mikroprocesor. Takva koncepcija se, razume se, nije mogla očekivati i od računara „galaksija", ali nju mogu (i, rekli bismo, moraju) da sprovedu svi oni koji nameravaju da naš računar koriste iole ozbiljnije. Iako su za hobiste, zbog oskudice na svetskom tržištu čipova, došli veoma teški dani, u redakciji smo razvili memorijsko proširenje „galaksija 48", koje „galaksiju" značajno približava „ozbiljnom" računarskom društvu. Proširenja omogućava povećavanja memorija u blokovima od po 16 K, a može se priključiti na verzija računara sa 2 i 6 K.



KOMPONENTE

Integrisana kola

TMS4416	2 4 III 6
74LS157	2
74LS08	1
74LS32	1 III 2
74159	1

OTPORNICI

R1 R2	220 oma
R3—R13	1—10 koma

Kondenzatori

C1	56 pF
C2	100 pF
C3—C7	100 nF
C8	10—47 uF (tantal)

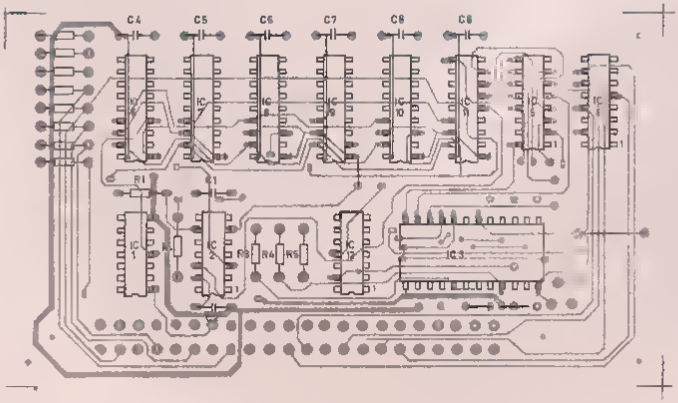
Podnožja

18 nožica	2, 4 III 6
16 nožica	2
14 nožica	2 III 3
24 nožica	1

Konektor

S × 22	razmak 1
3.95 mm	

Električna šema proširenja „galaksija 48" memorija podeljena na nezavisna polja od po 16 K



Montažna shema proširenja „galaksija 48" Iako je primenjena dvostrana štampa nakoliko kratkospojnika se nije moglo izbeći kratkospojnici koji prolaze ispod memorijskih čipova služe da skrate vezu kondenzatora za blokadu sa masom čime se povećava njihova efikasnost u peglanju napajanja za dinamičke čipova kratkospojnici ispod i iznad čipa 74159 služe za programiranje adresnog dekodera — na slici je prikazana situacija kada se u „galaksiji" nalazi 6 K RAM-a (kada su ugrađena sva tri memorijska čipa)

od spoljašnjeg hardvera predstavlja, praktično jedinu kompilkaciju u radu sa dinamičkim memorijama. Ova kompilkacija je međutim, prilično trivijalna i nije vredna pažnje ni sa stanovišta cene ni sa stanovišta složenosti, ni sa stanovišta pouzdanosti rada uređaja. Dinamičke memorije, uz to, rade dosiovce u milionima primeraka kućnih računara — nema nikakvog razloga zašto na bi mogle i u „galaksiji"

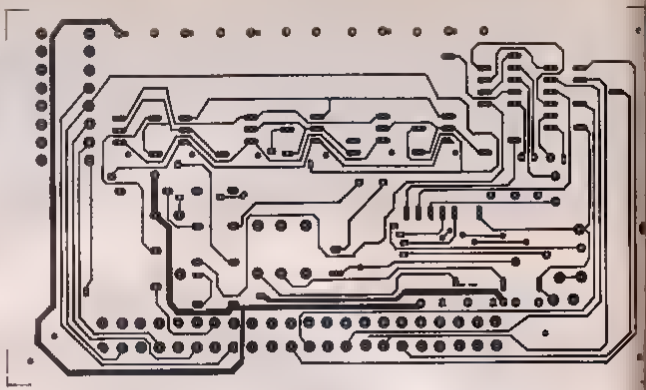
I sve i nešto

Na tržištu postoji četiri tipa dinamičkih memorija. Prvi među njima 4416 organizacija 16384 bita×1, predstavlja najjeftiniji i možda najlogičniji izbor. Osam čipova koliko je potrebno za 16 K, koštaju svega oko 2000 dinara. Ova memorije na žalost zahtevaju trostruko napajanje, a za veće kapacitete atrahovito složeno štampano kolo i mnoštvo — čipova! Po svojoj konfiguraciji i tehničkim rešenjima, ova memorija predstavljaju prevazidani tip. Memorija dvostruko većeg kapaciteta 32768×1 koje sa ugrađuju u računar ZX Spectrum, mogu da budu znatno privlačniji izbor. Ovaj tip memorija — dugo je trabalo da sa to sazna! — na proizvodi sa nigde u svetu. Radi se o memorijama 4164 (65536×1) kojima nije ispravna jedna sekcija!

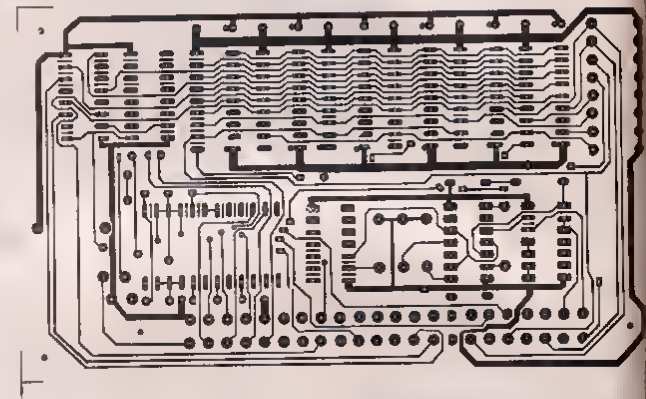
Tako, praktično, za izbor ostaju samo dva tipa. Osam čipova tipa 4164, za četrdesetak funti obezbeđuju memoriju kapaciteta 64 K! Iako se u „galaksiji" kao ni u jednom računaru ne može direktno iskoristiti čitav adresni prostor (nešto se mora ostaviti i za ROM, meda RAM pod određenim uslovima, može da radi i ispod ROM-a) ova konfiguracija predstavlja najatraktivnija rešenja. Memorija jednostavno pokriva sve prazna prostora u memorijskoj mapi, uključujući i prostor za ROM 2 i uspešno zamenjuje jedan ili dva statička RAM-a (D i E). Oni vlasnici „galaksija" koji još nisu nabavili ova čipove mogu, tako, za isti novac da kupe gotovo pola memorijskog proširanja.

Za ljubitelja potpunih rešenja — a potpuna rešenja su uvek i elegantnija i jeftinija od polovičnih — ove varijanta predstavlja jedini mogući izbor. U redakciji je razvijan prototip ovog rešanja koji sa odlično pokazao i koji će dobro poslužiti za razvoj sistamskog softvera, ali smo se ipak odlučili za drugu varijantu. Zašto? Ovo rašenja ima jedan dosta ozbiljan nedostatak filozofsko-finansijske prirode — ono pred vlasnika „galaksija" postavlja dilemu: sva ili ništa. Plašimo se da bi većina ljubitelja računara pre bila prisiljana da izabera ništa" nego „sve". Za njih bi to bila „supa za ždrala" — lepo ali skupo i nepristupačno rašenja.

Četvrti tip memorija 4416, organizacija 16384×4 bita, možde nije toliko fleksibilan u popunjavanju rupa u memorijskoj mapi, ali objedinjuja prednost potpunih i polovičnih rešenja. Memorija se, takođe, može proširiti do punog formata, ali sa to na mora uraditi odjednom — moguće je i u tripul u blokovima od po 16 K. Jedan ovakav blok koga čina dva čipa 4416 nije čak ni danas skupiji od jednog jedinog čipa 6116! Ovaj tip memorija je relativno nov i malo se leže nalazi na tržištu — drže ga



gornje strana



donje strane

Štampano kolo. Iako objavljujemo crteže obe strane štampanog kola, u razmeri 1:1 preporučujemo vlasnicima „galaksije" da se strpe do septembra i sačekaju fabričku verziju

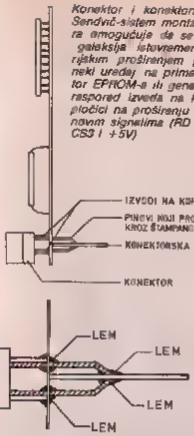
praktično jedino u Engleskoj — ali tako nače ostati zadugo.

Memorijski čip 4416 ima osam adresnih ulaza: A0—A7 i četiri ulaze za podatke: D0—D3 što praktično znači da jedan čip može da pamti samo pola bajta (četiri bita). Zato se za svaku polovinu bajta koristi po jedan čip. Preko kontrolnih ulaza **WE** (upisivanja/čitanja) **RAS** (upisivanja adrese za redove i osvežavanje) **CAS** (upisivanja adrese za kolona i izbor čipa) i **G** (dozvola izlaza) upravlja se procedurom upisivanja i iščitavanja podataka. Čip može da radi u nekoliko različitih modaliteta i veoma složenim memorijskim shamama. Mi smo razume se, odabrali najjednostavniji način rada.

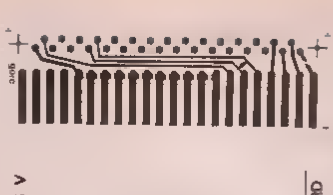
„Dinamička" procedura

Električna shama memorija „galakaja 48" izvedana je klasično za ovakvu vrstu sklopova, bez ikakvih ekstravagantnih zahvata i kompilkovanih vremenskih kola. Adrese za redove i kolone obezbeđuju dva multipleksera tipa 74LS157. U zavisnosti od stanja na kontrolnom ulazu **S**, multiplekser na izlaze Y postavlja ili adresu koju dobija preko ulaža A (A8—A13) ili adresu koja dobija preko ulaža B (A0—A7). Na početku ciklusa upisivanja ili čitanja multiplekser postavlja na adresna ulaza A0—A7 memorijskog čipa adresu za redove, a stotinak nanosekundi kasnije na ulaza A1—A6 adresu za kolone. Adresa za redove upisuje se u memorijski čip silaznim ivicom impulsa **RAS** (row address select — izbor adrese za radove) a adresa za kolona silaznom ivicom impulsa **CAS** (column address select — izbor adrese za kolona). Stotinak nanosekundi nakon kompletiranja adrese aktiviraju se izlazni baferi i podatak može da se pročita.

Procedurom upisivanja adresa upravlja se signalom **MREQ** preko nekoliko **ILI** kapija sa odgovarajućim kolima za kašnjenje. Na adresnim ulazima memorijskog čipa stalno se nalazi adresa za redove i oni se pozivaju svaki put kad se mikroprocesor signalom **MREQ** obraća memoriji — bez obzira da li želi da upiše podatak da ga pročita ili da osveži memorijska ćelija. Ako želi da upiše ili da pročita **WR** signal će pripremiti memorijski čip a **RD** ili **WR** otvoriti ILi kapiju N6. Pošlo je već upisao čip adresu za redove **MREQ** će, nako



Konektor i konektorska pločica Sendvič-sistem montaže konektora omogućuje da se na računar galaksija istovremeno sa memorijskim proširenjem priključi još neki uređaj na primer programator EPROM-a ili generator zvuka raspored izveda na konektorskoj pločici na proširenju dopunjen je nazivi signalima (RD CS1 CS2 i CS3 i +5V)



Konektorska pločica na računaru „galaksija" Da bi se omogućilo jednostavnije povezivanje periferijskih uređaja preko porta za proširenja konektorska pločica mora da pretrpi izvesne izmene sve izmene se vrše sa donje strane pločice a za njih su dovoljni skalpel lemilica i dva parčeta žice

izvesnog kešnjenja (R1 C1 N6), prekopčati multiplekser i dovesti ne A1—A6 adresu za kolone Nakon što se stabilizuje ova adresa ne izlazu kole za kašnjenje oko N7 (R2, C2) će se pojaviti signal **CAS** i upisati se u memorijski čip Prilikom upisivanje podataka čipu se istovremeno sa adresom za kolone saopšteva i podetek (tzv early write modaliet — upisivenje impulsom **CAS**) Prilikom čitenje podetak se pojavljuje spotinek nenosekundi nakon upisivanja adrese za kolone

Signal **CAS**, istovremeno upravije i glazno izlaznim baterime — podetak se ne može ni upisati ni pročitati eko ulaz kojim se upravlje nije ne niskom nivou To praktično znači da se on može koristiti i za izbor određenog memorijskog bloka Adresni dekoder 74159 određuje preme kom će bloku ovaj signal biti usmeren Dekoder pokriva čitav memorijski prostor u blokovima od po 4 K Pošto se redi o modelu sa 32 otvorenim kolektorima izlazi se mogu spojati paralelno bez ikakvog ograničenje Za blok od 16 K moreju se spojiti četiri Shema proširenja „galaksije 48 dopušte dve mogućnosti — nadovezivanje na postojeća 2 K redne memorije (spojiti tačke A—B C—D i E—F) i 6 K (B—C D—E i G) Šta će urade oni koji su u „galaksiji ugradili srednju varijantu od 4K? Moraju da dokupe još jednu statičku memoriju od 16 ili da presecanjem vode **CS** koji vodi prema memoriji D isključe ovaj memorijski blo U prvom slučaju memorije se može proširiti do 64K e u drugom samo do 56 K.

Upotrebljeni dekoder se nekome može učiti previše glomazan za ovakvu primenu To je međutim bio jedini način da se sve signe izvesne tleksibilnost sisteme Od toga osim toga mogu imati koristi čak i oni koji ne nameravaju da proširuju memoriju do kraja — na konektor su izvučeni signali **CS** za poslednjih 12 odnosno 16 K koji se elegantno bez potrebe za novim dekoderom mogu iskoristiti u nekom drugom proširenju — na primer proširenju ROM-e u kombinacijame 1×16 K 2×8 K ili 4×4 K! Oni koji se unapred odriču posleonjih 16 K redne memorije treba na štampanom kolu da preseku veze izmedu nožica 14 i 15 i 15 i 16 i time razdvoje **CS** signele u poslednjih 16 K adresnog prostore

Štampano kolo. . .

Štampano kolo je zbog dosta složene strukture uređeno u dvostrukoj štampi To će znamo razljutiti one hobiste koji sve vole da rade sami neročito eko na ono što i sami mogu da urade moreju da čekaju dva-tri meseca Ovakvi eklopovi, međutim, podrazumeveju šlempeno kolo profesionalne izrede — Inače se može desiti da nikede ne prorade Za one koji imaju razumevenje za domeće prilike i znaju da se neke stveri moreju i čekati, pripremili smo servis za nabavku štampanog kola Za one koji gore od nestrpljenja i koji imeju više poverenje u sebe objavljujemo crtež obe strane u razmeri 1 1 Sa jedne na drugu stranu veza se prenosi u svim tačkama koje se nalaze na gornjoj streni — najčešće preko pinove na integrisanim kolima eli ima i posebnih prenosnike U tim tačkama (treba li to reći?) elemenat se iemi sa obe strene Oni koji se odluče za samogradnju štempanog kole morače da nabave specijelne pinove za štempena kole poput onih na našim totogratijama ili de zaleme čipove direktno Ovo naravno važi samo za štampane kola iz amaterska redionice Na štampenom kolu koje obezbeđuje redakcije sve rupice će biti metalizirane i hobisti neće moreti da brinu gde se veze prenosi e gde ne Na ovekvom štempanom kolu lemi se naravno samo se donje strane

. . . i štampani konektor

Ne štempenom konektoru ne računeru galaksija nisu iz neobješnjivih razloga izvedena dva priključke bez kojih memorijsko proširenje ne može de radi — priključak ze +5 V ze napajenja proširenja iz računare i **RD** priključak sa signelom za iščitevenje memorije Zato se pre nego što se proširenje priključi na konektoru moreju izvršiti izvesne dorade nepejenje komedom žice treba dovesti ne priključak 1 koji je inače prazan e **RD** signel, direktno sa nožica 21 ne mikroprocesoru, na priključak 22, ovaj priključak je inače vezan se masom računera I ta se veza ne konektorskoj pločici mora preseći Da bi se oslobodio prostor ze **CS** signale treba, takođe, preseći i sve veze Izmedu priključka 8 i 15 Sa zadnje strane memorijskog proširanje postavljene je konektorska pločica Ona omogućuje da na računar, istovremeno sa proširenjem, buda priključen još jedan dodetek — na primjer, programetor epproma generetor tonova, Interfejs za štampeč Iii — nešto treće

Memorijski bagovi

Testiranje segređenog proširenje je krajnje jednostavno — ovo testiranje, zaprevo, „galeksije obavija eutomatski prilikom eveka inicijalizecije — i razuitat je vidljiv čim se uključi računar Dovoljno je dekle, otkucati naredbu PRINT MEM, koje izdaje količinu slobodne memorije, i ne ekrenu će se u zavisnosti od broje ugrađenih blokova pojeviti brojka od koje će se konstruktoru zavrteti u glavi. Na žalost, nije tako Ma koliko to čudno zvučalo operativni sistem „galeksije ne obezbeđuje potpunu podršku za memorijsko proširenje — u njemu sa potkrala nekolika bagova koji u izvesnim eituacijma oležavaju rad sa spoljašnjom memorijom Najozbiljniji bag sa odnosi na pun memorijski tormet. Umesto de u takvoj eituecij računa se 54 K slobodne memorije, koliko stvarno i ima. „galeksije u promenljivu „krej memorije" upisuje nulu i ponaša se kao da nema ni jednog jedinog slobodnog bajte — odbije da radi u programskom modelitetu! Kako je to moguće? Prilikom testiranje memorije, „galeksija postevija promenljivu „kraj memorije' na prvu lokaciju sa koje nema odziva — umesto na posledniu sa koje ga ima — e prva lokacije posle FFFF je 0000!

Ova teškoća će zagorčevati život samo onima koji na postojećih 6 K dograde još 46 K radna memorije. Rešenje je jednostavno eli neselegantno i dosta zamorno eko se pre početka rede u promenljivu „kraj memorije (&2A6A) upiše FFFO , geleksije će ponovo postati poslušne do sledećeg — uključenje Ručno postavljanje tzv „ramtopa predstavlja u ovoj eltuacij, na žalost jedino rešenje U svim ostaiim situacijema „krej memorije se poetavlja kao treba, eli to praktično, znači de se moramo odreći čitavog memorijskog bloka (kako „odseći' samo šesnaest bajtova?) koji, u našem siučaju iznosi 4 K.

Naredba PRINT MEM de stvar bude teža sa više memorije počinje de izdaje eva čudnije rezultate One ispravno funkcionišе samo do 22 K redne memorije, odnoeno do adrese 32767. Nekon toga daje negativan broj iz kog se ne može izvući eme baš nikakav logičen zeključak o stvarnoj količini slobodne mamorije Sa proširanjem od 16 K PRINT MEM daje rezultate 21445, od 32 K — 27706, e od 48 K — 11322! Ispitivenje promenljive „krej mamorije — PRINT WORD (&2A6A) — daje takođe negativna brojeve — 32768 (16 K), — 16384 (32 K) i 0 (48 K) U prostoru iznad 32 K RAM-a „galeksije u naredbame tipa BYTE i WORD prihveta samo heksadekadne eli ne i decimalne pozitivne brojeve Sve ova ograničenja, međutim nemaju nikakvog uticaje na uobičajeni rad sa memorijom — ona potpuno normalno prihvete i bejzik i mašinske programe

Projekat i tekst: Jova Regasek
Crteži: Mire Tiodorović

Yuqoslavia Commerce

SONY

SONY® Yugoslavia Commerce

YUGOSLAVIA COMMERCE je preko 20 godina ekskluzivni zastupnik SONY-a za našu zemlju. Predmet zastupništva je celokupni proizvodni program SONY-a iz oblasti Hi-Fi i video opreme kao i kompletni program U-MATIC profesionalne opreme. Kompletan asortiman je obezbeđen u Beogradu Sarajevska broj 5.

Od marta ove godine u Sarajevskoj br.3 otvoren je izložbeni salon SONY-a. Za sada salon nije kompletiran ali će do kraja godine u njemu svoju konačnu dislokaciju. Trenutno je izložena Hi-Fi i video oprema. To je jedna prva faza na putu pronalaženja stalne koncepcije izložbenog salona.

U drugoj fazi koju očekujemo krajem maja biće izložena U-MATIC video oprema. U-MATIC oprema biće posebno interesantna za RO iz oblasti turističke privrede, mada je njena primena daleko šira u bankarstvu u osiguranju društvu, institucijama... Na primeru turističke RO objasnićemo vam jedan od načina primene U-MATIC video opreme. U-MATIC je pogodan

— Za animiranje i prezentiranje kompletne turističke ponude na turističkim berzama i sajmovima. Poslovnim partnerima moćićete na najatraktivniji i najdinamiku turističku sezonu. Jokom ove godine Vaše mogućnosti nisu više ograničene sezonom.

— U turističkim agencijama pogotovo u onim koje se bave plasmanom turističkih aranžmana i prodajom izleta. Obogatite vašu ponudu video spotovima. Rezultati sigurno neće izostati.

— Za projekciju filmova muzičkih sportskih i drugih emisija. Ponudite vašim goalima video. oni su na to već navikli. Možda im baš to odgovara. Loše vreme na sme da vas iznenadi.

— Za organizovanje interne televizije.

U trećoj fazi ustanovljavanja SONY-avog salona biće predstavljano uvođenje kompjutera i video sistem. Ova faza očekuje se krajem novembra 84 godine. Kompletiranjem ove tri faze izložbeni salon mogao bi da postane ogledni primer jedne banke podataka o trenutno najsavremenijoj audio video i kompjuterskoj opremi.

Ako vam naš salon nije usputan, posetite nas na jednom od sledećih sajmova

— Sajam tehnike u Beogradu od 22 — 26 V 84 god.

— Sajam elektronike u Ljubljani od 3 — 10 X 84 god.

— Interdisk u Zagrebu 16 — 24 IV 84 god. Da vas ukratko upoznamo sa našim poslovnim partnerom Kompanijom SONY.

Malo je firmi u svetu koja su popul SONY-a, za tako kratko vreme. Izmanila naše svakodnevne navike. Još je manja firmi koja su to uradila tako munjevito.

All krenimo od samog početka.

Firma SONY osnovana je u maju 1946 godine. MAJARU IBUKA i AKIRO MORITA sa početnim kapitalom od samo 500 (USA) dolara osnivaju malu firmu-radionicu. TOKYO TELECOMUNICATION ENGINEERING CORPORATION. jednu od mnogobrojnih koja su nastale odmah posle II svetskog rata, u Japanu. Za nepuna četiri decenije SONY je pravalio put od male radionice do svetskog giganta. U prvoj deceniji lansiran je prvi japanski tranzistor (po američkoj licenci) a ubrzo prvi u svetu SONY izbacuje tranzistorski radio prijemnik. Tako je započela SONY-eva svetska ekspanzija. Kasnije sve je bilo jednostavnije. Pokrenuta je lavina pronalazaka koji će lagano osvajati fabrike, institute domove, Ostalo su učinili sposobni inženjeri-elektroničari i sprelni menadžeri.

SONY svake godine lansira na svetsko tržište nove proizvode koji postaju sastavni deo života savremenog čoveka. Kao ilustraciju nabrojaćemo samo neke SONY-eve proizvode

— porad spomenutog tranzistorskog radio prijemnika za SONY je vezana i pojava prvog tranzistorskog TV prijemnika

— 1958 prvi tranzistorski video rekordar

— 1969 prvi tranzistorski portabl video rekordar

— 1964 prvi kućni video rekordar

— 1969 razvijen U-MATIC sistem (3/4 inča)

— 1975 razvijen BETAMAX sistem (1/2 inča)

— 1976 razvijen OMEGA sistem (1 inč).

Pored ovog ne možemo a da ne spomenemo

— TV sistem trinitron

— usavršavanja video i audio traka (Fe i Cr traka su SONY-ev izum)

— WALKMAN (personalni Hi-Fi)

— MAVICU (prvi elektronski fotoaparat).

— COMPAKT DISK (prvi digitalni gramofon u saradnji sa PHILIPS-om)

— BETAMUVIE (prva video-kamera rekorder)

— WATCHMAN (prvi džepni televizor)

— BETA Hi-Fi (prvi video rekorder sa Hi-Fi zvukom)

SONY-ev izuzetan doprinos razvoju Hi-Fi i video tehnike ovim nije iscrpljen. Spomenimo i trenutno najmanju kameru na svetu SONY CCD-65. Očekuje se njena svetska premijera.

Organizacionu strukturu kompanije SONY čine tri njena dela.

Prvi deo koji je i bio osnovan za razvoj ostala dva radi na razvoju oprema široke potrošnje. Tu se proizvodi Hi-Fi i video oprema.

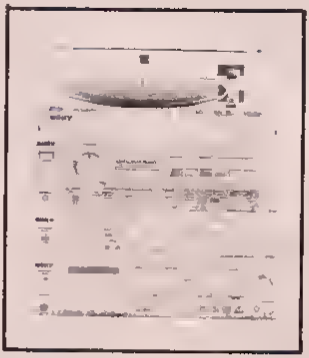
Drugi deo osnovan 1965. godine je takozvani INSTITUTIONAL VIDEO baziran na sistemu U-MATIC.

I na kraju treći deo firme SONY-BRODKAST nastao je tak sedamdesetih godina razvojem video-rekordera od 1 inča.

SONY-evi ciljevi do 1990. godine su da sa zadrži pozicija vodeća video kompanija u svetskim razmerama. Strateško težište je na razvoju novih medija kao što su telefaksat videoteks), kablovska televizija satelitska televizija.

U svojoj orijentaciji SONY nije zapostavio ni razvoj mikrokompjutera. Krajem godine biće predstavljen SONY-ev personalni kompjuter (uglavnom za poslovnu primenu) sa kompletnom periferno opremom i u ovoj deceniji može se očekivati dinamičan razvoj. posebno ako se zna da SONY ulaže 5-6% godišnjeg profita u istraživanja i osvajanja novih proizvoda.







Yugoslavia Commerce
OOUR YUCOM TRADE
Knaza MILOŠA 60/11000 Beograd
011/689 166

IZLOŽBENI SALON SONY®
11000 Beograd
Sarajevska br. 3
011/646-344

svi spektrumovi restarti

Umetnost programiranja

Pri mašinskom programiranju procesora Z80 pozivanje potprograma se vrši instrukcijom CALL. No ovaj procesor obezbeđuje još jednu instrukciju kojom se može izazvati izvršenja određenog potprograma. To je tzv. restart (RST). Prednost ove instrukcije je što u programu zauzima svega 1 bajt (u poređenju sa CALL — 3 bajta) pa samim tim zahteva manja vremena za njeno izvršenje. Ali ovom instrukcijom je moguće pozvati samo potprogram koji počinje na jednoj od sledećih adresa: 0/H0 /8H 1/H0 1gh 28H 28H 30H ili 38H. Odmah primećujemo da se ove adrese odnose na najniži memorijski prostor i to na tzv. nultu stranicu (zero page) memorije, tačnije prvih 256 bajta. Pogled na memorijsku mapu Spektruma uvariće nas da u ovom računaru svih 8 restart adresa pripadaju ROM-u tj. ne njime počinju neki potprogrami operativnog sistema. Naravno postoje izmedu samih restart adresa je suviše mali za iole ozbiljniji potprogram, pa obično ovi potprogrami započinju skokom ili pozivom glavnog dela potprograma koji se nalazi na nekom drugom mestu u memoriji.

Pri kreiranju jednog složenog operativnog sistema kao što je u Spektrumu vodi se računa o tome da se na restart adresama postave potprogrami koji se u toku rada najčešće pozivaju. To omogućave da se u glavnom delu operativnog sistema vrlo jednostavno, trošeći jedan bajt memorije, dakle RST instrukcijom pozove neki od tih potprograma. Zašto ovaj luksuz ne bismo ponekad priuštili i sebi pišući neki svoj mašinski program?

Pre svega moramo upoznati tačnu funkciju svakog od ovih restart potprograma. Mi ne možemo menjati postojeći program u ROM-u (osim ako neko nije raspoložen da piše sopstveni operativni sistem) pa se stoga ukoliko želimo da koristimo njegove potprograme, moramo prilagoditi njegovim zubicama.

Pogledajmo stoga redom šta i kako rade pojedini restart potprogrami u Spektrumu.

RST0: počni od početka

Poziv ovog dela operativnog sistema na poseći nigde u ROM-u. Razlog je jednostavan: po uključenju računara, procesor započinje izvršavanje instrukcija upravo od adrese 0 — posle provara RAM-a i inicijalizacija sistema prestaje es radom dajući na ekranu svima poznatu copyright poruku. To znači da bi instrukcija RST 0 u nekom programu imala apsolutno isti efekat kao da smo isključili i ponovo uključili računar. Stoga ova instrukcija ima smisla jedino ako je upotrebimo na kraju nekog programa a želimo da on bude uništen i sistem reinicijalizovan.

RST 8: prijavi grešku

Potprogram koji započinje na adresi 08 ima puna ruka posla dok preko tastatura unosite neki bejzik program. On je naime zadužen da izvrši normalizaciju steka i nekih sistemskih promenljivih svaki put kada interpreter naiđe na grešku u vašem programu. Osim toga on na pogodan način doturi interpreteru kôd greške tako da se kasnije ona i tekstualno prikaže.

U čemu je značaj normalizacije steka? Ukoliko pogledamo bilo koji složeniji mašinski program vidimo da se on sastoji od manje ili više potprograma koji se pozivaju iz jednog glavnog dela. Ovi potprogrami mogu takoda dalje da pozivaju svoje potprograme itd. Ako radimo mašinski program koji recimo briše ekran ili tome slično nećemo imati potrebe da prijavljujemo greške. Međutim ako pišamo program koji treba da prepozna određena komande ili koji radi interaktivno tj. zahteve od korisnika unošenje nekih podataka moramo kontrolisati te podatke pri unosu i javljati određenom porukom da je došlo do greške jer recimo sintakse ili podaci ne odgovaraju. Šta da radimo ako greška nastupi u nekom potprogramu koji poziva drugi potprogram a njega opet poziva pretpostavimo glavni program? Pri svekom uzastopnom pozivu na stek se ostavljaju povratne adrese i jednostavnom naredbom za povratak iz potprograme (RET) ne možemo se tek tako lako vratiti tamo gde hoćemo. Ako vršimo direktan skok (JP ili JR) u potprogram za raguliasanje greške posle sveke greške imaćemo sve više podataka na steku (ostavljenih usled CALL instrukcija) i on će se neprosto zagušiti a program krahirati.

Svih ovih i nekih drugih briga može nas vrlo elegantno osloboditi instrukcija RST 08. Ona će pozvati rutinu iz operativnog sistema koja će vratiti stek u prvobitno stanje i izvršiti indirektan skok u neš potprogram koji će dalje raguliasati grešku. Naravno mi moramo na neki način javiti ROM-u gde se te naše rutine nalazi. To činimo tako što u sistemskoj promenljivu ERR SP (23613) ubacujemo recimo na početku rada našeg programa adresu na kojoj se nalazi neš potprogram za prijavljivanje grašaka. Najjednostavniji način da se ovo uradi je da sa na početku programa na stek ubaci adresa naše rutine i sadržaj tek pointera (SP) odmah sačuva u ERR SP. To bi onda izgledalo oveko

```
LD HL, ERRENT     adresa rutine za greška
PUSH HL           idi na stek
LD (ERR SP), SP   sačuvaj stek pointer
```

Nakon ovoga dovoljno je da posle detekcije greške u programu jednostavno izvršimo RST 08 i dalji rad se nastevlja u našem potprogramu za prijavljivanje grašaka se stekom ne prvobitnoj adresi.

Ako u programu treba da prijavljujemo više različitih grašaka, onda ize instrukcije RST 08 postavimo jeden bajt koji će predstavljati kôd određene greške. Posle izvršenja ovaj kôd će se zehvaljujući potprogramu iz ROM-a neći u sistemskoj promenljivoj ERR NR (23610). Naša rutina za greške će tada jednostavno preuzeti ovaj kôd proanalizirati ga i prijaviti odgovarajuću grešku. Veome je vežno da po izvršenju ovog potprograme ili ne njegovom početku vratimo na stek njegovu adresu (PUSH instrukcijom) bi RST 08 i dalje mogao funkcionisati ispravno. Još nešto ako hoćemo da po izvršenju ma... skog programa nastavimo rad u bejzik sistemu konstimo RST 08 na opisani način onda ... ikasvog postavljanja u ERR SP treba sadržaj ... sistemske promenljive negde sačuvati (ima ... zgodno mesto od dve bajte u zoni sistems... promenljivih na adresi 23728) a pra povratka bejzik vratiti joj originalni sadržaj. Ukoliko ... dirali ERR NR pre povratka u bejzik sistem t... de u ovu lokeciju ubacimo vrednost 255 (0FF...

Za sve ove probleme, naravno postoje mnoga druga rešenja ali je verovatno najbo... koristiti potprograme iz ROM-a kad god je moguća. To će pre svega smanjiti veli... našeg programa čemu uvek traba težiti.

RST 10: ispiši karakter

Od svih ovih restart potprograma u RO... programeru će verovatno najviše koristiti o... koji započinje na adresi 10H. Njegova funkcija je da izvrši ispisivanja znaka, doturanog u akum... toru procesora na ekran ili štampač. Pod ... kom se ovde podrazumevaju ne samo alfanu... rički i grafički karakteri već i svi kontrolni ko... koje obezbeđuje Spectrumov set karaktera. ... potprogram radi nešto sporije upravo zbog t... što obuhvata sve ove znakove. Stoga ako za... vamo maksimalno brzo ispisivanje na ek... recimo u igrama, onde nem jedino preostaje ... pišemo svoja rutine za ispisivanje. No ako vi... ispisivanja nije kritično onde će nas RST ... rešiti dosta problema, neročito oko Spectru... nezgodne video memorije.

Kao što smo već rekli da bi smo ispisali ... karakter potrebno je da u akumulator atav... ASCII kôd karaktera i izvršimo RST 10H. Al... praksi stvari zahteveju malo više pažnje i pr... mnog rada. Pre svega, pre bilo kakvog ispis... nja ROM-u traba javiti gde vrlimo ispis ... glavni deo ekrana, donja dva linije ekrana ... štampač. To činimo tako što u akumulator u... mo 2 za glavni deo ekrene 1 za donji deo ili 3 ... ispis na štampač te zatim pozovemo potprog... na adresi 1601H. Ovo je potrabno uraditi de bi ... zoni kanalskih intormacija (channel informa... u RAM-u postavila odredene izlazna adresa ... ispisivanje. Nakon toge ispisivanje će se ... tamo gde smo odradili sve dok ponovnim ... vom rutine na adresi 1601H ne promanimo ... ječe stanje.

Kao što je već rečeno na raspolegerju u ... nam svi kontrolni kodovi (AT TAB kodov... boja pozicija kursora itd.) koko je predstavl... u uputstvu ze Spectruma na strani 183. ... kontrolnih kodove za INK i PAPER, traba da ... kôd boje (0—7) a iza FLASH, INVERSE, BRI... i OVER 1 ili 0 zavisno od toga da li uključ... mo ili isključujemo odredenu funkciju. Iza ... i TAB moramo dati dve parametra za AT isto ... no kao u bejziku dok TAB kontrola zahteva ... sledeći podetek bude stvarni TAB parametar ... je drugi podatak nebitan ali ga moramo ... slati.

Mašinsko programiranje na računaru ne može se ni zamisliti bez dobrog poznavanja operativnog sistema. Većina rešenja koja su ljubiteju računara potrebna čak i za najjednostavniji program već se nalazi u ROM-u — na programeru je samo da ih „pronađe" i iskoristi na svoj način. Teško je naći kućni računar koga prati tako obimna, raznovrsna i detaljna dokumentacija kao ZX Spectrum. Pa ipak, njegov operativni sistem je još uvek pun tajni i one koji se spremaju da zarone u njega sasvim sigurno očekuju prijatna iznenađenja. Za prikaz Spectrumovog ROM-a i opis svih rutina u njemu teško da bi bilo dovoljno i čitavo ovo izdanje. Zato prikaz ograničavamo na osam najvažnijih rutina, koje u ROM-u zauzimaju sasvim posebno mesto i koje računar u toku rada najčešće proziva.

Zbog greške u ovom potprogramu ROM-a naročito moramo obratiti pažnju da ne dozvolimo ispis kontrolnog koda 8 (cursor left) u momentu kada su koordinate za ispis podešene na 0/0 (gornji levi ugao ekrana) jer će sa desiti vrlo čudne stvari autoru ovog teksta se dešavalo da bejzik program buda oštećen a moguć je čak i krah programa. Na ostalim pozicijama nema nikakvih problema. Kao i za svaki pojedini znak za svaki kontrolni kôd i njegove parametre potreban je poseban RST 10H. Naravno, u programu ćemo to ostvariti pomoću petlji ili sličnih rešenja.

Za one koji su zainteresovani za „iziazak" iz ROM-a ovde se pruža šansa za tako nešto. Pošto RST10H pre ulaska u samu rutinu za ispis karaktera traži njenu adresu u RAM-u mi bismo mogli na to mesto u RAM-u (channel information) podvaliti adresu naše rutine i na taj način preuzeti celu kontrolu. Teško bi bilo sasvim moguće na mesto neiskorišćenih kontrolnih kodova dodati neke nove na primer za brisanje ekrana automatsku tabulaciju i slično. Posle provere kôda ukoliko nije naš izvršili bi smo skok u rutinu u ROM-u (počinje na 09F4H) na odgovarajuće mesto pa bi ona dalje dovršila posao. Ovo je sasvim izvodljivo ako pozivamo RST 10H iz našeg mašinskog programa. Stvari postaju bitno komplikovanije ako hoćemo da na RST 10H iskačemo u našu rutinu za vreme normalnog rada bejzik interpretara. Razlog za to je što jedne od rutina za brisanje ekrana u ROM-u direktno postavlja u RAM adresu potprograma za ispis (09F4H). Znači ako mi izmenimo adresu na koju preiazi RST 10H, već prvo brisanje ekrana uništiće našu adresu. Protiv ovoga se donekle može „boriti" tehnikom interapta o čemu će ako bude interesovanja biti reči drugom prilikom.

RST 18 i RST 20: uzmi sledeći karakter

Dve kratka potprograma na adresama 18H i 20H bejzik interpreter intenzivno koristi za vreme izvršavanja bejzik programa. Funkcija im je da srodno sistemskoj promenljivoj CH ADD (23645)) u kojoj se čuva adresa do koje je interpreter stigao analizirajući liniju bejzik programa vrate interpretaru kôd karaktera koji je adresiran preko CH ADD (RST 18H) odnosno od prvog sledećeg karaktera (RST 20H). Evidentne primene ovih restart rutina bi mogla biti u mašinskim programima koji na neki način nadograđuju bejzik interpretar ili za eventualna proširenja postojećeg bajzika u Spectrumu.

RST 28: počni da računaš

Na adresi 28H počinje Spectrumov kalkulator kao što je poznato on omogućuje rad sa pokretnim zarezom i obezbeđuje Spectrumu sasvim prietojan set matematičkih funkcija. Rutine kalkulatora zauzimaju dobar deo ROM-a no uobičajeno je da se poziv kalkulatora vrši instrukcijom RST 28H iza ove instrukcije treba da sledi sukcesivno bajti koji će odrediti kalkulatoru šta da radi sve do bajta (38H) koji određuje kraj rada kalkulatora. Izvršavanje programa se dalje nastavlja od sledeće instrukcije.

Sva funkcije koje su na raspolaganju u bejziku možemo upotrebiti i u našim mašinskim programima. Dobro upoznavanje ovog kalkulatora može biti od ogromne koristi svima koji su zainteresovani za rešavanje složenijih matematičkih problema u mašinskim programima. Kalkulator raspolaže sa 66 operativnih funkcija i 5 konstanti. Izračunavanje nekog matematičkog problema se svodi na to da se on razbije na takve delove koji odgovaraju operativnim funkcijama kalkulatora i zatim reši u sukcesivnom izvršavanju ovih funkcija. Table i objašnjenja u vezi ovih funkcija i konstanti bi nam odnele previše prostora u ovom tekstu te preporučujem čitaocima koji su zainteresovani za ove probleme sledeće knjige „Understanding your Spectrum" od Dr Iana Logane i The Complete Spectrum Rom Disassembly od istog autora.

RST 30: napravi malo mesta

Namena ovog kratkog potprograma koji se inače vrlo često poziva iz operativnog sistema jeste da u određenom prostoru u RAM-u (tzv. work space) napravi mesta za onoliko bajta koliko mu se doturi u BC paru registara. S obzirom da funkcionalne preko odradenih sistemskih promenljivih čini mi se da ovaj potprpgram za sada nema naku upotrebnu vrednost.

RST 38: prekini i očitaj tastaturu

Restart potprogram na adresi 38H se bitno razlikuja od svih do sada spomenutih specifičnost lokacije 38H je u tome što procesor ukoliko mu ja naređeno da radi u režimu interapta (interrupt — prekid) i to u modu 1 (IM1) svakih 20 milisekundi (dakle 50 puta u sekundi) prekida izvršavanje programa zepamti gde je stao i preiszi upravo na sdresu 38H da odatle izvrši potprogram koji na njoj započinje. Kada ga završi vraća se na glavni program i to tačno na mesto na kom ja bio prekinut. Iz ovoga proizilazi da u stvari rada dve programa paralsino glavni koji interpretira bejzik itd i drugi na adresi 38H. Postavlja se pitanje šta to radi taj potprogram kad mu je već dodeljena čast da radi uporedo a bejzik interpraterom? Odgovor je očitave tastaturu i pokreće Spectrumov interni časovnik. No svakako glavni zadatak mu je da očita tastaturu i u sistemsku promenljivu LASTK (23560) ubeći kod tipke koja je eventualno pritisnuta. Očitavanje testature sa na vrši pomoću ovog potprograma za vrama izvršavanja bajzik naredbi LOAD SAVE i BEEP zbog isključenja interapta- mada SAVE i LOAD imaju svoje instrukcije za direktno očitavanje tipke BREAK.

Što se tiče naših mašinskih programa koji koriste tastaturu sva dok na naredimo isključenje interapta instrukcijom DI (disable interrupts), u sistemskoj promanljivoj LASTK ćemo dobijati kod pritisnuta tipka uz prethodnu proveru kontrolnog bita u sistemskoj promenljivoj FLAGS (23611) — ukoliko je bit 5 jednak nuli nikakva tipka nije pritisnuta a ako je jedansak jedinici onda možemo da preuzmama kod tipka iz sistemske promenljive LASTK. Jedno takvo tipično očitavanje bi izgledato ovako

```
WAIT HALT sačekaj interapt
BIT 5 (IY+1) tipka pritisnuta? (IY=5C3AH)
JR Z WAIT; ne — skok nazad
RES 5 (IY+1) mala predostrožnost
LD A (LASTK) uzmi kod tipka
RET izlez
```

(Naredba HALT sa može izostaviti. Labela WAIT tada ide na BIT 5 (IY+1).)

Ako nam je za rad programa neophodno isključanje interapta (naredbom DI) moraćemo sami da očitavamo tastaturu. Ukoliko se radi o igri gde sa očitavaju 3—4 tipke to ćemo uraditi direktnim očitavanjem sa porta pomoću IN instrukcija. Lepo objašnjenje ovog postupka je dato u knjizi „Spectrum mechina code for absolute beginner". Međutim ako nam je potrebna kompletna testatura možemo pozivati kad je to potrebno potprogram iz ROM-a na adresi 028EH ili još zgodnije sa RST 38H a tim da odmah sledi naredbe DI pošto pri izlezu iz rutine na adresr 38H postoji naredba EI (Enabie interrupts) za omogućanje prekida. Posle ovakvog očitavanje kôd eventualno pritisnuta tipke biće posta vijen u LASTK a bit 5 u FLAGS će javiti de ja tipke zaista pritisnuta. Evo primera očitavanje testature u tom slučaju

```
READ RES 5 (IY+1) mala predostrožnost
(IY=5C3AH)
CALL 028FH, poziv potprograma iz ROM-a
BIT 5 (IY+1) dali je neke tipka pritisnuta?
JR Z READ ne — skok nazad
LD A (LASTK) uzmi kod tipka
RET izlaz
```

CALL 028FH može biti zamanjen se RST 38H a zatim DI.

U slučaju da pišemo program koji radi u režimu interapta u modu 2 možemo u servisnu rutinu koja se izvršava po nastupu interapta ubeciti RST 38H i ne taj način imati očitavanje testature — kao u modu 1 (normalan rad bejzik interpretara i operativnog sisteme) pošto interapt u modu 2 nastupa u istim vremenskim intarvalima kao u modu 1 (svakih 20 ms).

Ovo bi bilo u kratkim crtama ono osnovno što moramo znati o nuttoj stranici Spectrumovog ROM-a i njegovim restart potprogramima de bi smo ih uspešno koristili u našim programima. Za detaljnije analize neophoden je dissembling ROM-a a od velike pomoći mogu biti i knjige na ovu temu već spomenute u tekatu.

Milan Pavićević

UNIS-RO promet birosredstava (p.o.)-Sarajevo

Olympia International — olivetti

INFORMATIONS UND KOMMUNIKATIONS-SYSTEME

ELEKTRONSKI SISTEMI ZA PISANJE



OLYMPIA INTERNATIONAL ESW 103
NOVA KLASA ŠTAMPAČA SA LEPEZOM (typewheel)

Gdje god su Vam potrebni pouzdanost i oštro-oblikovana kucana strana ESW 103 je alternativa optimalng odnosa cena/performanse.

Pisaća mašina visokih performansi — naravno sa kolaturnim mehanizmom — za korespondenciju. Ili savršeni štampač za Vaš kompjuterski sistem. ESW 103 poseduje obe funkcija. Selekcija između tipove operacija, bilo kao pisaća mašina ili kao štampač, izvršava se jednostavnim pritiskom na taster.

Bilo standardno pisanje različitim stilovima, proporcionalno pisanje masna slova i/ili prostorno pisanje od kojih svaki ima svoj znak, svaki ima kvalitet koji odgovara Vašem imidžu.

Intelegencija štampača oslobađa Vaš kompjuter suvišnog opterećivanja. U tazi rada na sistemu, primljeni tekst štampan je maksimalnom brzinom i najbržim metodom. Lampe na tastaturi su i pomoć rukovaocu i simultano ukazuju na odgovarajući operacioni mod.

ELECTRONSKA PISAĆA MAŠINA OLYMPIA ESW 103

Tehnički podaci

Tip mašine: Tip 103 je mikroprocesorom kontrolisana output pisaća mašina sa interface-om. Dodatno funkcijama pisaće maši na ova mašina može biti kao output ili input/output uređaj u on-line modalitetu. Ukupno 3 interface-a su na raspolaganju za oba, jednostrani ili dvostrani tok podataka.

Kontrolna elektronika *Kontrolna elektronika sastoji se od tri pod-montaže*
1 Kontrolor štampača
2 Kontrolor modaliteta
3 Interface

Interface (kako izaberete)
IEC bus (slušač)
IEEE 488/ANSI MC 1.1
Centronics compatabilni, 7(8) bits
EIA RS 232 C (CCITT V 24)
TTY 20 ma current 100p 50-19 200 Bauda

Snaodjevanje energijom (kako izaberete)

220V	50Hz	110/220V	60Hz
240V	50Hz	110 127V	60Hz
110/220V	50Hz	120V	60Hz
100V	50Hz	100V	60Hz

Sekundarno UMG=+36V
UMK=+9V
Preko regulatora voltaža UDD=+5V

Osigurači Primarni 1 termički osigurač (115 st. C)
Sekundarni 1×1.6 amp sporo-eksplodirajući
1×4.0 amp sporo-eksplodirajući

Potrošnja struje U radu — 70 watti
U pripremi — 35 watti

Sigurnosna stipulacija * Odgovara Evropskoj sigurnosnoj stipulaciji BSI VDE SEV, NEMKO itd.
** Odgovara internacionalnoj sigurnosnoj stipulaciji npr. IEC 380
*** Odgovara prekomorskoj sigurnosnoj stipulaciji CSA UL

Štampač Štampač sa lepezom (daisy wheel) 96 znakova, pozicioniranje štampača i lepeze pomoću koraka motora. Bizine štampanja 17 znakova/sekunda

Metoda štampanja Automatsko napr.jed/nazad automatsko potiskivanja funkcije među prostora

Stilovi štampanja. Masni tip, prostorno kucanje, dupli udari

Razmak slova (pitch) Standardni stilovi 1/10", 1/12", 1/15" proporcionalno pismo 1/60", osnovno kucanje 0.42 mm

Dužina linija:
141 znakova sa razmakom 10 (pitch)
169 znakova sa razmakom 12
212 znakova sa razmakom 15

Data konstrukcija: ASCII standard i sekvenca izbegavanja za kontrolu štampača

Mehanizam traka za pisanje Specijalna kaseta
Ručni izbor priprema indigo i karbon trake, automatski izbor multi-udarne karbon trake štampanje u dva nivoa.
- Standardna karbon traka, lift-off karbon traka dužina 130 m, širina 13 mm, broj udara 100 000

CENTRALNI DEO VASE KA

Multi-karbon traka, dužina 115 m širine 13 mm broj udara 328 000
Tkanana traka crna dužina 9 m širine 13 mm broj udara 1 080 000
Tkanana trake crno/crvene dužina 9 m širina 13 mm broj udara 270 000

Tastature Tastatura sa 48 tastera tip TUMBLER Svi tasteri sa funkcijom ponavljanja osim tastera ekscenata Tasteri funkcije povrat štampača sa i bez preleza na sledeći red prostorni taster polukorak, povrat za jedno mesto, ispravka taster za pomeranje i fiksno pomeranje.
Kontrole desnog dela tastatura
tasteri za desnu i levu marginu, taster ON/OFF--LINE tastar tabulatore prostor među redovima i taster za razmak (pitch) slova, tasteri za napr ed/nazad vertikalno pozicioniranje valjka polukorak, preko funkcije ponevljanja
Kontrole levog dela tastature
oslobađenje margine, taster ta bulatora (tabuliranje una zad), taster šifre (co de), tester pona vljanja
Kontrole u kombinaciji sa tasterom šifre: ma sna slova, prostorno pisanje, dupli udar

Indikator tempe: ON/OFF-LINE greška masna slova, prostorno pi sanje dupli udar

Veljak: Dimenzije veljke dužina 43,5 cm prečnik 44.5 mm

Buffer: 2K i 4K byte buffer podetaka u ON-LINE operacijama

Memorija isprevkt: 256 byte (1 linije) u OFF-LINE operacijama

ELEKTRONSKI SISTEM ZA PISANJE ETS 1010

Ukoliko već posedujete elektronsku pisaću mašinu Olivetti već ste načinili prvi korak u elektronsk kancelarijski sistem Olivetti Vašu elektronsku pisaću mašinu možete koristiti sa esvremenim uređajem sa ekrenom
ETS 1010 ima CRT displej sa 25 linije dva pogonske diskete kapaciteta sa 360 K-bajte memorije i mogućnošću povezrvanja se ETS 121

Za upis mformacije jednostavno otkucati tekst (tastatura ima konvencionalni izgled) Informacije se prikazuje na displeju (U ovoj fazi tekst na mašini se ne kuca) e kede se završi upis ili izmena teksta štampe se konačan tekst brzinom od 20 znakova u sekundi Ono što vidite na ekranu-doterani tekst, masni slog i podvučena slova sada dobijete otkucano na papiru

Lako je naučiti i raditi se ETS 1010.

ETS 1010 olakšave prelaz na upotrebu automatike uključivanjem u elektronsku pisaću mašinu OLIVETTI i održava bliski i komandnu vezu sa korisnikom Mašina je još uvek najelikasnije sredstvo za pisanje kratkih obavesti, indekanih kartica ovrata itd

ETS koristr tehniku lagenog pomaranje tekste Kada se na papiru pomera jeksl ne vrše se preskoci od jednog reda u odredenom vremenskom periodu Umesto toga kreće se polako i omogućava jednostavno čitanje Da bi se olakšala upotreba ETS ima funkcije UNDO i HELP Ukoliko nešto redite (npr umatanje paragrafe ne pogrešnom mestu) e zatim uočite grešku nemojte brinuti jednostavno poništite to (UNDO) ETS 1010 će sva dovesti u red

Ukoliko niste sigurni u vezi sa funkci jom ili šte treba de ije da redite tražite po moć (HELP) 1010 će vas provestr kroz sve funkcije ETS 1010 lakode poseduje široki opseg novih funkcije za pomoć u upravljenju

Vanserijska funkcije.

Jedinatvena sposobnost vizuelnog sistema u ETS 1010 predstevlja novu standerdnu odliku za leksu obra du fusnote, naslova i brojeva kao predmeteka kao i za prenošenje informacije od jednog dokumente na drugi
Vizuelni sistem omogućava de se ekran podeli na dva ili više površina za protok informacije Veći dokumenti (duži od 60 kolona ekrene) se obrađuju u mnogim sistemima horizontalnim pomeranjem Jedini problem je u tome što se gubi tekst Upotrebom kapeciteta prosledivanja u sistemu ETS 1010, leva kolona se može „zamrznuti", dok sa preostali deo dokumente kreće

ETS 1010 pomoć u edmtnietretivnom rukovodstvu.

ETS 1010 poseduje „ledica za dokumentaciju" i „registre" za veće dokumente Možete staviti dokument u erhivu ili jednostavno referencu prema položaju dokumenta Tehniku vođenja dokumentecije je leko razumeti jer ona obezbeduje brzu dostupnost ilsli svih ledica za dokumantaciju, regtatrima i dokumentima kao i obaveštenje o datumu nastanka autoru, daktilografu i određenom ključu identifikacioni sletem za automataku revizlju štedi vreme edmlnistrirenje time što ubrzave korekturu Ukoliko je potrebno obaviti ovu funkciju ETS 1010 može emitoveti ili štampati izbrisane reči precrtene linijom ili dodete reči sa crtom iznad njih
Sa ostalim venserijskim elementima ETS 1010 će uskoro pomageti u proračunima planiranju budžete i periodučnim proračunrne i pripremt izveštaja.

ETS 1010 ze pouzdanoet

Kvelitet i pouzdenost su oznake izvedbe ETS 1010 ETS 1010 može podneti preženjenje etetičkog elektriciteta koji se često javlja zbog suve klima i zagrajenih prostorije Ukoliko je pad napone ili nestanak struje u rednim prostorijama čest, za zaštitu od uništenja informacije služi baterije (po izboru kupca)



SVE DETALJNE INFORMACIJE KOD ZASTUPNIKA ZA JUGOSLAVIJU

UNIS

UNIS-RO promet birosredstava (p.o.)-Sarajevo

UDRUŽENA METALNA INDUSTRIJA
SARAJEVO

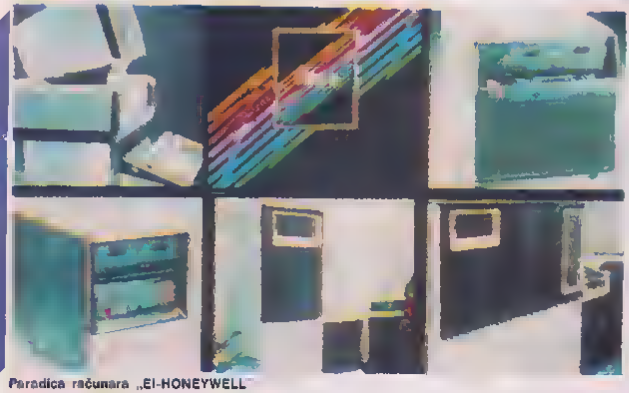
Đure Đakovića 45 d, 71000 Sarajevo
telefoni 071/ 516 355
39 342

CELARIJE BUDUCNOSTI

Honeywell

INFORMACIONI SISTEMI

PROGRES



Paradiса računara „EI-HONEYWELL"

„EI-HONEYWELL" NUDI OPREMU PO POTREBAMA I MOGUCNOSTIMA KORISNIKA, ZATO, POTRAŽIMO RESENJE ZAJEDNO!



Računski centar sa sistemom „EI-HONEYWELL" DPS 6/95 Karakteristike tog modela su

— najnapredniji model iz serije EI-HONEYWELL DPS6
— 32 — bitni procesor visokih performansi za komercijalna i naučne obrade
— memorija kapaciteta 2 MB do 16 MB
— mogućnost priključenja do 112 terminalskih radnih jedinica i/ili komunikacionih linija
— kapacitet diskova do 3,5 GB (giga bajta)

EI ELEKTRONSKA INDUSTRIJA NIŠ
RO „EI-RAČUNARI"
OOUR „EI-HONEYWELL INFORMACIONI SISTEMI"
18000 NIŠ, Bul. Nelke Vučetića 60-62
tel. 018/292-342, 234-088
telex: 18000 YU EI-HIB

Honeywell
INFORMACIONI SISTEMI
SEKTOR MARKETINGA
11000 BEOGRAD, Terazije 3/4
tel. 011/643-444, 322-436
telex: 11537 YU EI-HIB

RO „PROGRES"
OOUR „PROGRES-INFORMATIKA"
11000 BEOGRAD, Maršala Tita 67/VIII
tel. 011/685-947, 641-655
telex: 11400 YU PROG
Predstavništvo:
41000 ZAGREB, Gundulićeva 4
tel. 041/447-679
61000 LJUBLJANA, Gregorčičeva 4
tel. 061/224-760

EI — HONEYWELL INFORMACIONI SISTEMI

„EI-HONEYWELL" Informacioni sistem je preduzeća sa zajedničkim ulaganjem (JOINT VENTURE) američke firme **„HONEYWELL INFORMATION SYSTEM" Inc.** sa sedištem u Mineapolisu, poznatog svetskog proizvođača računara **ELEKTRONSKE INDUSTRIJE",** Niš jednog od najpoznatijih preduzeća u oblasti elektronike i digitalne tehnike u Jugoslaviji i **„PROGRESA", OOUR "PROGRES-INFORMATIKA, Beograd,** kao generalnog zastupnika i distributera opreme HONEYWELL, za koju pruža korisnicima svu stručnu pomoć, i kao takvo preduzeće bazirano na uspešnoj saradnji 3 partnera, obavlja sledeće delatnosti

— **Proizvodnju** DPS6) modela mini srednjih i velikih računara, video-terminala i serijskih štampača iz proizvodnog programa Honeywell-a
— **Plasman** mini, srednjih i velikih računara na jugoslovanskom tržištu
— **Export** mini, srednjih i velikih računara u više od 50 zemalja sveta posebno u nesvrstane zemlje i zemlje u razvoju
— **Projektovanje** informacionih sistema
— **Razvoj** aplikativnih paketa
— **Distribuciju** sistemskog softvera i aplikativnih paketa
— **Instaliranje,** održavanje i poslove field-engineerig-a
— **Obuku** korisnika računarskih sistema

Svi računari se proizvode pod znakom **„EI-HONEYWELL",** koji garantuje kvalitet, proizvoda na svetskom nivou sa automatski prenetim novim znanjima i otkrićima samog „HONEYWELL -a

Danas na 5-godišnjicu potpisivanja JOINT VENTURE Ugovora, možemo se pohvaliti da smo sa otvaranjem nove fabrike u Nišu (1 maj 1981 uspeli da zadovoljimo velike potrebe jugoslovanskog tržišta, i dosada plasiramo preko 300 sistema koji su dobili mnoge pohvale i priznanja od strane zadovoljnih korisnika

NOVO U IZDANJU
IRO »VUK KARADZIĆ«

ELEKTRONSKI RAČUNAR

Most u budućnost

SVE ŠTO STE HTELI DA ZNATE O RAČUNARIMA, PROČITAJTE U OVOJ KNJIZI

Elektronska obrada podataka, kao polje aktivnosti savremenog čoveka, i elektronski računar, kao tehnološki proizvod koji omogućava tu aktivnost, poslednjih godina se izuzetno brzo razvijaju i šire. U svakodnevnom životu su sve prisutniji i imaju sve veću ulogu.

Literatura iz ove oblasti je u svetu veoma brojna i bogata. Kod nas, na žalost, to nije slučaj. Postoji samo manji broj udžbenika i priručnika okrenutih prvenstveno programskim jezicima i teorijskim osnovama računara namenjenih studentima. Nedostaju tekstovi koji daju celovit prikaz računara i njegovih mogućnosti na popularan, iako dosledno stručan način dostupan svima čije je poznavanje matematike i tehnike na elementarnom nivou.

OVA KNJIGA TREBALO BI DA POPUNI OVU PRAZNINU



Latinica,

130 strana,

format. 165×22cm,

60 fotografija i ilustracija,

tvrd, plastificirani povez.

Cena: 900 din.

Izdavačka radna organizacija „Vuk Karadžić"
Beograd Kraljevića Marka 9 tel 620-371

NARUDŽBENICA RAČUNARI II

Neopozivo naručujem knjigu ELEKTRONSKI RAČUNAR po ceni od 900 dinara ZA GOTOVO — sa popustom od 20% Plaćanje pošiljaoca prilikom prijema knjige NA OTPLATU u 2 mesečne rate. Rate ću otplaćivati redovno svakog meseca sa prijemnu knjige računa i odgovarajućeg iznosa uplatnice „Vuk Karadžić" Kamata da mi se obračunavati prema broju rata i važeći m zakonskim propisima

(Nepotrebno precrtati)

U slučaju spora priznajem nadležnost Opšt nekog suda u Beogradu

Prezime očevo me i ime.....

Mesto godina rođenja

Zanimanje——————Tačt broj i mesto stanovanja——————

Ulica i br oj telefon——————

Naziv adresa radne organizacije...

telefon——————

Ovom e zapečaćenju za kupca na otplatu Potpis kupca

Br i karte izdate od SUP-a——————

Penzioneri prilažu preispodeđenju ček od penzije

Iskra Delta

računalniški sistemi delta







RO ISKRA DELTA je proizvođač kompletnih računarskih sistema sa uhodanim razvojem i proizvodnjom mašinska opreme, sistemske i aplikativne programske oprema, sa organizovanom prodajom, školovanjem, održavanjem i inžanjeringom.

Osnovna koncepcija RO ISKRA DELTA temelji sa na proizvodnji računarskih sistema i periferniih jadinica sa maksimalnom primanom domaće tehnologija i znanje, a na osnovu najnovijih svatskih dostignuća u ovoj obiasti.

Iskra Delta
proizvodnja računarskih
sistema i inženjering

LJUBLJANA, Parmova 41
Telefon 061/312-988 h.c.
Telex 31366 YU DELTA